Exploring multi-granular documentation strategies for the representation, discovery and use of geographic information

James Kenneth Batcheller

Thesis submitted for examination for

Doctorate in Philosophy

in

Geographical Information Science

Department of Geography

The University of Edinburgh

2009

Declaration

I declare that this dissertation represents my own work and that where the work of others is used it is duly accredited. I further declare that this dissertation consists of four research papers, a shorter research article, an introductory chapter and a concluding chapter, with a combined length of approximately forty thousand words.

James K Batcheller

September 2009

Acknowledgements

I would like to thank my supervisors Femke Reitsma and Bruce Gittings for their heroic patience, continued guidance, considerable feedback and valuable contribution over the course of producing this thesis, without whom it would not have been possible. Mention is due to Rob Dunfey for his help with ArcObjects and feedback on paper IV, as well as to the GIgateway Advisory Group for graciously allowing me to attend a number of their meetings. I would like to thank Cameron Easton and his team in the Scottish Government for entrusting me to develop a small but hopefully not insignificant piece of the 'One Scotland – One Geography' initiative, the GI Project Registry, something that has since been successfully developed to address what shortcomings I left it with. I would also like to express my sincere gratitude to Steve Dowers, not only for his support with all things IT but also for acting as a tolerant sounding-board for what were mostly quite inane research-related ideas. Thanks go to Arthur for providing the rewards I so infrequently deserve. I am also grateful to the Deringer clan, not least for supplying regular consignments of hearty foodstuffs on which I have dined over the last four years. For their continued support since earlier than I can remember I remain indebted to my family, Batchellers, Harringtons and Muellers, with particular mention for my mother Rosemary and father Edgar Hadley. Finally but most of all, for putting up with me over the course of this work through good times and rough, I must thank my better part Natalia, and our daughter Elisa Rose. Your unwaivering support has meant more to me than I can possibly express and I dedicate this thesis to you both.

Abstract

This thesis explores how digital representations of geography and Geographic Information (GI) may be described, and how these descriptions facilitate the use of the resources they depict. More specifically, it critically examines existing geospatial documentation practices and aims to identify opportunities for refinement therein, whether when used to signpost those data assets documented, for managing and maintaining information assets, or to assist in resource interpretation and discrimination. Documentation of GI can therefore facilitate its utilisation; it can be reasonably expected that by refining documentation practices, GI hold the potential for being better exploited. The underpinning theme connecting the individual papers of the thesis is one of multi-granular documentation. GI may be recorded at varying degrees of granularity, and yet traditional documentation efforts have predominantly focussed on a solitary level (that of the geospatial data layer). Developing documentation practices to account for other granularities permits the description of GI at different levels of detail and can further assist in realising its potential through better discovery, interpretation and use. One of the aims of the current work is to establish the merit of such multi-granular practices. Over the course of four research papers and a short research article, proprietary as well as open source software approaches are accordingly presented and provide proof-of-concept and conceptual solutions that aim to enhance GI utilisation through improved documentation practices. Presented in the context of an existing body of research, the proposed approaches focus on the technological infrastructure supporting data discovery, the automation of documentation processes and the implications of describing geospatial information resources of varying granularity. Each paper successively contributes to

the notion that geospatial resources are potentially better exploited when documentation practices account for the multi-granular aspects of GI, and the varying ways in which such documentation may be used. In establishing the merit of multi-granular documentation, it is nevertheless recognised in the current work that instituting a comprehensive documentation strategy at several granularities may be unrealistic for some geospatial applications. Pragmatically, the level of effort required would be excessive, making universal adoption impractical. Considering however the ever-expanding volumes of geospatial data gathered and the demand for ways of managing and maintaining the usefulness of potentially unwieldy repositories, improved documentation practices are required. A system of hierarchical documentation, of self-documenting information, would provide for information discovery and retrieval from such expanding resource pools at multiple granularities, improve the accessibility of GI and ultimately, its utilisation.

Table Of Contents

List Of Figures

INTRODUCTION

PAPER I

PAPER III

PAPER IV

DISCUSSION AND CONCLUSIONS

List Of Tables

Introduction

Geographical Information (GI) relates to the physical location of features, objects and phenomena relative to the surface of the earth and may involve maps, addresses, Global Positioning Systems (GPS), remote sensing images and related technologies. The importance of GI can be inferred from both the copious volumes produced and its application across a host of diverse domains, a minute subset of which include coastal zone and flood defence management, pollution monitoring, crime and pestilence analyses, urban planning, emergency services and vehicle routing, utilities management, and so on. Combined with the resource-intensive nature of generating GI, means for enhancing its efficient use have become increasingly relevant.

Documenting GI contributes to its use by assisting in its discovery, discrimination and management through the accessible depiction of key properties that might otherwise be unreadable, or not formally recorded. Whether employed by organisations internally to safeguard information assets or used to publicise and uncover GI through the use of associated discovery services, documentation systems have been used for assorted purposes successfully for some time. Nevertheless, this thesis poses the question whether existing documentation approaches can not be further enhanced, with the ultimate goal of enhancing the efficient use of GI. More specifically, what are the options for improving existing documentation systems, to better enable access to the GI they describe? Can the processes surrounding documentation be augmented to mitigate the effort necessary to create it, and in turn, increase the likelihood that such documentation exists? What options exist for documenting GI at varying detail, and what are the benefits and implications for doing so?

The current work attempts to address these questions over the course of four research papers and a shorter research article, connected by the overall theme of multi-granular GI documentation (as illustrated in the hierarchy presented in Figure 1). Paper I deals with geospatial *dataset metadata* and how they are employed in spatial data clearinghouses to facilitate data discovery and exchange. Presented in the context of the UK's public sector initiative GIgateway, the metadata creation to publication lifecycle is reviewed and enhancements to its function in facilitating data access suggested. Paper II (short research article) describes GI *project documentation* and presents the GI Project's Registry (GIPR), an early deliverable of the Scottish Government's, "One Geography – One Scotland" GI Strategy. Together, the GIPR and project documentation are proposed as tools for better informing geospatial data pursuit, as means for raising the profile and awareness of GI-related activity and for promoting collaborative opportunities across Scotland. Paper III presents a first attempt at automating geospatial dataset (or *layer*) metadata. It outlines a means of supporting manual metadata record authoring through the computation and collection of contributory metadata elements from a layer's ambient computing environment, all in an effort to mitigate a frequently implicated obstacle to geospatial data exploitation. Paper IV continues to develop this work, sourcing further contributory elements for automating layer-level metadata. The input of *feature-level metadata*, conventionally used to document the contents of geospatial vector layer contents, is established via a method that aggregates such metadata for inclusion in that of the layer-level. The role of geometry in automating documentation is also developed, as is the completion of narrative metadata elements that may otherwise be overlooked. Finally, Paper V presents an approach for implementing feature-level semantics and how they may be used to support the discovery and use of data within a geospatial web application. In a departure from the syntactic strategies pursued in the other papers of the thesis, the work uses formalised semantic tools to document geospatial resources in a way that can account for varying data perspectives.

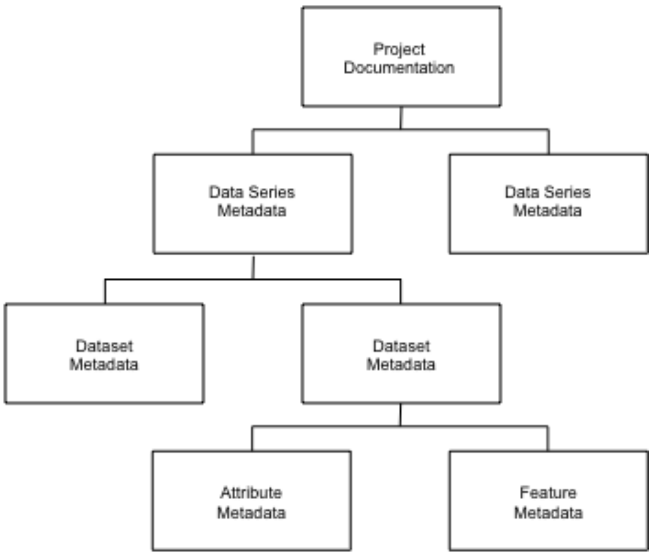


Figure 1. The metadata hierarchy employed for documenting GI at varying granularity. Data series metadata are not extensively addressed in the present work but nevertheless have a role in describing collections of datasets otherwise depicted individually. Feature-level metadata are represented by attribute and feature metadata.

Documentation

The American Heritage Dictionary of the English Language[1] generically defines documentation as the "collation, synopsizing, and coding of printed material for

[1] Dictionary.com, "documentation," in The American Heritage® Dictionary of the English Language,

future reference". Its more specific definition for the computer science domain reads as the "organized collection of records that describe the structure, purpose, operation, maintenance, and data requirements for a computer program, operating system, or hardware device". The use of the word in the current work, as with that of the wider GI community, draws upon aspects of both definitions and entails the methods or instances of the description of geospatial resources[2] for varied purposes such as their discovery, discrimination, interpretation and access.

Geospatial Data

The main focus of the present research is on the category of GI commonly used within Geographic Information Systems (GIS), i.e. digital geographic data. More specifically, GIS or *geospatial*[3] data will be used to generally refer to the vector representation of geographic point, line or polygon features (and their attributes) unless indicated otherwise. These conditions are merely intended to narrow the degrees of freedom of the current work, not indicate a limited relevance of the work presented. Approaches elaborated for geospatial data as defined above could in certain circumstances be applied in documenting physical cartographic products or digital aerial photographs for instance.

Fourth Edition. Source location: Houghton Mifflin Company, 2004. http://dictionary.reference.com/browse/documentation. Available: http://dictionary.reference.com. Accessed: March 27, 2009.

[2] While much of the thesis deals with data as the geospatial resource in question, the importance of the geospatial service as a resource is not overlooked.

[3] The terms spatial and geospatial have been used interchangeably throughout the current work; the formal definition of geospatial (or geo-spatial) data as shorthand for geo-referenced spatial data is nevertheless acknowledged.

Geospatial Metadata

Geospatial metadata are used to document geospatial data. Just as geospatial data represent real world abstractions, designed for applications ranging from cartographic representation to spatial analysis, geospatial metadata represent similar abstractions, albeit of the data themselves. Metadata are often characterised as 'data that describe data' (Hart *et al.* 2001, Tsou 2002, Hobona *et al.* 2004); here this is further specified as the key properties that describe a specified unit of (geographic) information. These properties may be used to detail elements such as the identification, spatial coverage and quality of geospatial data, narrative abstracts describing the data, as well as how they may be accessed and exchanged (Kim 1999, Limbach *et al.* 2004).

Geospatial Metadata Standards

Geospatial metadata standards have underpinned modern geospatial data documentation efforts, with a particular emphasis on their role in facilitating data discovery, exchange and interoperability. Depicting metadata elements with a common vocabulary facilitates their interpretation; content standards provide a "common set of terminology and definitions for the documentation of digital (geospatial) data" (FGDC 1998). Digitally capturing metadata elements in a predictable manner facilitates their manipulation by both man and machine; encoding standards aim to specify this structure. Used together, content and encoding standards allow metadata to be recorded consistently, collected within metadata catalogues, indexed and made available for discovery.

Geospatial layers

Geospatial data are commonly instantiated as *layers*, also known as a datasets or feature classes. Layers are employed by GIS software to represent features of a particular geometric type (point, line, polygon) for a given geographic extent (e.g. region, country, continent). The location of the world's capitals, the course of rivers in Africa or the extent of administrative boundaries of the UK – each would for example be manifested within their own distinct layer. Layers offer a logical construct most frequently used for modelling, representing, analysing and transferring vector GIS data. Unsurprisingly, geospatial metadata approaches have long exhibited a layer-centric bias, often recording data descriptions in standards-compliant text-based eXtensible Mark-up Language (XML) documents (Batcheller 2008).

Paper I – The data clearinghouse model

The first paper of this thesis deals with one such layer-based strategy, the data clearinghouse model. As focussed online dataset registers or as components of broader Spatial Data Infrastructures (SDI), clearinghouses provide mechanisms for cataloguing distributed collections of data layers by means of standardised metadata (Nebert 2004). Specific elements are extracted from each metadata record into an index; it is against this index that remote users search for data layers, on the basis of keywords and geographic extent for example. Layer metadata therefore assume a central role in the operation of clearinghouses; this chapter explores this role, and reviews some of the challenges clearinghouse contributors face in managing the metadata creation to publication lifecycle. With the UK's GIgateway service as a reference implementation, the various transitions layer metadata undergo from creation to dissemination are reviewed, impediments to these transitions are identified and mitigating strategies presented. Options for developing the core clearinghouse model are also discussed, the primary focus being on how layer metadata may be leveraged in moving beyond the discovery of datasets and towards their delivery.

The GIgateway service provided a noteworthy case study, representing the UK's flagship geospatial data discovery service for the public sector. GIgateway arose from foundations laid by the National Geospatial Data Framework (NGDF), the initial vision of which was as a fully functional SDI of national scope (i.e. NSDI). Concerns over the ownership of the service, its strategic goals and sources of revenue, the promotion of the service and who constituted the target community all contributed to the decision to scale back this original vision. Following a number of consultations with the GIgateway Advisory Group, it was clear that these complications have persisted to the detriment of the service. Further, in light of the European Union directives[4] that formalise requirements for the provision of public sector GI, coupled with the ongoing evolution of other Internet-based geospatial data delivery services, it can be reasoned that expectations have evolved beyond what a focussed discovery service can offer (Batcheller and Gittings 2006b). These circumstances have combined to ensure that GIgateway faces a somewhat uncertain future; Paper I represents a timely overview of the service, as well as options for enhancing it.

---

[4] Namely Public Sector Information (PSI, 2003) and the Infrastructure for Spatial Information in Europe (INSPIRE, 2004).

Paper II – Project documentation (short article)

Clearinghouses have an inherent constraint when used to pursue geospatial data layers, in that the searches they facilitate remain centred on layer-level metadata. Layer-level metadata may be adequate in depicting associated data and facilitating their discovery, but they convey little of the organisational or socio-political context in which datasets are created and curated (Comber *et al.* 2007). Differing perspectives can sway how geography is modelled for a given area, particularly where there are contending motives behind data collection and collation. What spurs data creation, who funded it, and other factors that influence the processes involved can for instance provide further criteria with which to evaluate layers and assist in discriminating between candidate datasets yielded in search result sets. Existing abstract fields of layer metadata records could be extended to incorporate project-level information, but this approach is less than ideal – such details would need to be proliferated across all layers produced or curated by the project, all captured within a field that should specifically describe its associated dataset[5].

Paper II is a brief discourse on project documentation and a live implementation known as the GI Project Registry (Batcheller and Gittings 2006a). Essentially a project-level metadata approach, the Registry accommodates the description of geospatial initiatives in what may be viewed as a companion application to data exchange schemes such as GIgateway. As the first tangible deliverable of the

[5] Abstract entries of data series metadata, or metadata that apply to a collection of data layers of similar type, objective or provenance, are similarly unsuitable, particularly in cases where there are multiple data series to a single project.

Scottish Government's "One Scotland – One Geography" strategy[6], the GI Project Registry offers an online, publicly accessible catalogue of geospatial initiatives from Scotland and beyond. Where data layer metadata can provide insight into the data they depict, the Registry's project documentation can be used to provide insight into the wider context in which data are generated and used, with the implication of better informing data gathering operations. On a broader scale, further benefit of such an approach can be seen in the manner in which GI initiatives and research activity can be showcased, enabling the identification and promotion of opportunities for improved engagement within the GI community.

Paper III – Automating data layer metadata

Of course, the performance of both project documentation and layer-based clearinghouses is dependent upon the availability of their respective metadata. Despite its proponents in industry, academia and the public sector, metadata are notoriously neglected, particularly when layer metadata and their generation are considered (Mathys 2004). In light of a curious lack of (published) pragmatic approaches for automating the creation of geospatial metadata to help mitigate this neglect, Paper III presents an initial attempt at a software solution designed to assist in the creation of layer metadata (Batcheller 2008).

The approach, informed in part by automation techniques developed in the library community, is founded upon the customisation of a proprietary GIS software application and a popular geodatabase format in which data layers are held. It is

[6] http://www.scotland.gov.uk/Publications/2005/08/31114408/44098

based upon the idea that by combining the processes of data layer preparation, filing and documentation into a consolidated workflow, obstacles to layer metadata generation can be lowered while enforcing a system of organisation that safeguards data. The presented prototype illustrates how metadata items of a chosen standard may be computed from: a layer's dataset construct through the extraction of configuration parameters set during data registration; a layer's attributes by indexing commonly occurring keywords and from the layer's ambient computing environment, such as from pre-compiled content templates, a systematic storage protocol and settings from the underlying operating system. The technique demonstrates that layer metadata automation is entirely feasible; it does not however suggest a dispensability of human actors in the generation process, least of all in the layer preparation and metadata mediation stages. Even so, simplifying the creation of metadata will make its availability more probable, and in turn, improve capitalisation of overlooked or underexploited datasets by facilitating their accelerated location, appraisal and retrieval.

Feature-level metadata

In facilitating this location, appraisal and retrieval, layer metadata by definition looks past the description of individual geographic features, focusing instead on the collection. Feature-specific details must therefore be generalised, which may lead to a loss of potentially useful information. Consider the example of a layer metadata element that registers the survey method associated with a roads layer as having been performed using high-precision equipment. If certain stretches of road in the layer are subsequently updated based off a low-resolution aerial image, a dilemma is posed as to what the survey method element should now represent. Both methods could be recorded in the layer metadata within a compound metadata element (i.e. elements that permit multiple values), although this would result in a loss of potentially valuable information: where revised application requirements demand high-precision data, the entire layer would need to be resurveyed, as opposed to a subset of low-precision features. There are therefore instances where a metadata approach, aimed at the feature level, is warranted.

Each geographic feature stored within a given vector dataset is comprised of a spatial and aspatial component: the geometry of the feature, and its attributes. Storage strategy will vary according to implementation (Batcheller *et al*. 2007), but conceptually a feature may be viewed as a row within a geospatial database (or *geodatabase*) table, a record comprised of geometry and attribute fields. Metadata for this granularity of data can be handled in one two principal ways. An *embedded* feature-level approach is one where metadata elements are encoded alongside the data themselves, occupying fields in each record like any other attribute[7]. An *associated* feature-level approach is one that involves registering metadata apart from the data, such as within an external file in a manner similar to layer metadata. ISO 19115 for instance suggests a strategy where only feature-specific anomalies are captured as feature-level metadata, but within an associated layer-level metadata document (ISO 2005). In the aforementioned scenario of survey method registration, this latter strategy would suffice – only low-precision features would require identification. Embedded metadata on the other hand, being tied to the data they

[7] This may also be referred to as record-level metadata

describe, is a more appropriate approach where subsets of features are accessed, manipulated or extracted apart from their host layer.

Feature metadata and attribute metadata

Feature-level metadata may be further deconstructed into feature metadata, applied to the entity's spatial component, and attribute metadata, applied to the entity's aspatial component. Upon initial scrutiny this distinction may appear pedantic; indeed, registering metadata at even the feature-level may be too intricate for certain applications considering the level of effort it can demand. Where this record-level metadata is necessary however, ambiguities emerge if this distinction remains undefined. Consider a cadastral geodatabase detailing parcel geometry and ownership information. A *Date of record update* field may appear to offer a reasonable device for tracking change at the feature-level, but it does not distinguish between feature events, such as changes to parcel geometry, and attribute events, such as changes of ownership. Teasing apart the semantics of geometry-specific information from that for attributes and associating metadata accordingly facilitates this distinction.

The difference between feature-level metadata approaches and (geo-)spatial data standards in general is often equally unclear. Spatial data standards are generally designed to impose a predictable structure and content on attribute schema, facilitating data interpretation and exchange (Arctur *et al.* 1998). Feature and attribute metadata are similar in this regard, and as such may be considered as specific types of spatial data standard. A problem posed by certain standards however, such as the Spatial Data Standards for Facilities, Infrastructure and Environment (SDSFIE) (Halfawy *et al.* 2006), is that both feature and attribute metadata are frequently entwined as previously described. Feature and attribute metadata can allow for more specific and concise documentation to circumvent such issues.

Paper IV – Further automating data layer metadata

The work outlined in Paper IV makes use of such a feature metadata approach to further develop the layer metadata automation approach of its preceding paper. Paper III suggests the use of indexing techniques for generating metadata items from semi-structured attribute data; Paper IV on the other hand proposes a formal schema to structure the storage of feature metadata, embedded with layer attribute data (Batcheller *et al.* 2007). Where the former yields keywords that require manual mediation prior to use, the predictable arrangement and nomenclature of feature-level metadata of the latter means that items may be more amenable to being aggregated automatically for immediate use as layer-level metadata elements.

Feature metadata aggregation aside, other prospects for layer metadata automation are investigated. Layer geometry, arguably the most significant and defining characteristic of a geospatial dataset, offers one under-exploited source of metadata-relevant information. Through the use of a reference layer depicting application-relevant boundaries, a method is outlined that extends the contribution of geometry to metadata creation beyond that of current practices that focus on the coordinate extents of layer features. Elsewhere, assisting descriptive metadata creation using a technique that seeds entries in a record's abstract element is elaborated, providing *aides-memoires* to aid manual completion. And in an attempt to improve layer metadata effectiveness in facilitating data evaluation and pursuit, a means of coupling asset visualisation with metadata creation and output is described.

Paper IV is also framed in a more contemporary context to that previous, and aims to highlight geospatial metadata's function in broader community initiatives (e.g. GIgateway) and the paradoxical neglect its automation receives. It continues the case that opportunities for automating data documentation exist, and that the energy expended in data preparation can be leveraged to contribute towards such efforts. While founded upon proprietary software, key aspects of the study are highlighted, drawing the attention to how they may inform a more open metadata generation strategy not bound to a particular proprietary application domain, storage format or operating environment. The realisation of such a generic solution would enable broader consumption, improve the likelihood that data are adequately documented and in turn, capable of being better exploited.

Paper V – Feature-level semantics

While data documentation can facilitate its exploitation, this is predicated upon the condition that the artefacts of documentation are themselves interpretable by man and when necessary, manipulable by machines. The forerunning papers of this thesis outline metadata approaches of varying granularity, all of which are founded upon syntactic specifications. In other words, they are grounded in the assumption that both technological and human actors share and can interpret a common vocabulary, one that is imposed through the use of an agreed standard. However, in an increasingly distributed computing environment where digital resources may be variously sourced, combined, processed and redistributed, there is no certainty that the data or their metadata will be interpreted unambiguously. Consequences of this misinterpretation can include the misapplication of data or their not being exploited to the fullest potential.

Paper V departs from the focus on the syntactic and introduces a feature-level metadata strategy based upon formalised semantic methods. Syntactic-based information management systems can be susceptible to problems of arising from content ambiguity; the geospatial domain is no different in this regard. For instance, where feature descriptions are intended to be synonymous, such as ‘road’ and ‘street’, instances of each will be treated as distinct entities. A search operation for ‘road’ will only return road features; streets will be omitted. Encoding the semantics of feature attributes and implementing computing logic that can process such semantics can counter potential ambiguity in attribute content. Further, it can offer fresh insights into the data themselves and improve interoperability across systems by allowing for various data perspectives, ultimately improving data utilisation.

The paper presents a method for incorporating formalised semantics within a custom location-based search application, built with open source software components. Attributes of the geographic features underpinning the application are imbued with meaning using the Resource Description Framework (RDF) data model[8]. In other words, features are marked-up with metadata statements that describe the feature itself (resource), an attribute (property) and the attribute value (object). An associated Web Ontology Language (OWL)[9] ontology is used to model the application domain, essentially acting as an applied vocabulary specifying the concepts used and the relationships that hold between them. The RDF mark-up and ontology are coupled with a semantic framework that enables both query of base assertions present in the source data as well as implicit associations that would otherwise remain obscured. Combined with a geodatabase back-end and a map-driven interface, a system architecture based upon feature-level metadata is presented that can handle content ambiguity and help maximise the effectiveness of geospatial layers exposed through the service.

Thesis outline

While focussing on geospatial resource documentation of varying granularity, and on the related geospatial metadata systems, the underlying premise of this thesis is that documentation practices as a whole are merely means to an end. Each paper, while designed to stand on its own merit, successively contributes to the view that documentation practices can ultimately help maximise geospatial data utilisation. A final discussion and conclusions section follows the main body of the work, aimed at collecting the key subjects raised in the thesis and discusses their potential implications for current and future geospatial information systems.

---

[8] http://www.w3.org/RDF/
[9] http://www.w3.org/TR/owl-features/

Contributors

Any shortcomings of the current work are solely attributable to the author; any successes are in no small part thanks to those individuals who have contributed to varying degrees in the completion of this work.

Paper I – Bruce Gittings offered the original view of the timeliness of a review of GIgateway, provided a sounding board for many of the ideas elaborated within the paper and reviewed and proof-read various drafts of the paper. Femke Reitsma provided further review and feedback.

Paper II – Bruce Gittings originated the idea of developing a service for highlighting GI-related projects in Scotland, and reviewed and proof-read the article published in GeoConnexions UK.

Paper III – no contributions received.

Paper IV – Bruce Gittings provided review and proof-reading services; Robert Dunfey assisted the author in commencing the development of the metadata utility in ESRI's ArcObjects.

Paper V – Femke Reistma assisted in developing the concept for the paper, provided review and proof-reading services.

References

Arctur, D., Hair, D., Timson, G., Martin, E. and Fegas, R., 1998. Issues and prospects for the next generation of the spatial data transfer standard (SDTS). *International Journal of Geographic Information Science*, 12 (4), 403-425.

Batcheller, J.K., 2008. Automating geospatial metadata generation—An integrated data management and documentation approach. *Computers and Geosciences*, 34 (4), 387-398.

Batcheller, J.K. and Gittings, B., 2006a. A GI Projects Registry for Scotland. *GeoConnexion UK*, 4 (4), 32-34.

Batcheller, J.K. and Gittings, B.M., 2006b. Avenues for developing the UK's National Geospatial Metadata Service. *In*: G. Priestnall and P. Alpin, eds. *Proceedings of the Geographical Information Science Research UK (GISRUK) 14th Annual Conference*, 5 - 7th April 2006. Nottingham, UK: University of Nottingham, 259–262.

Batcheller, J.K., Reitsma, F. and Gittings, B., 2007. Implementing ISO-compliant Feature Metadata. *In*: A.C. Winstanley, ed. *Proceedings of the Geographical Information Science Research UK Conference (GISRUK) 15th Annual Conference*, 11-13th April 2007. Maynooth, Republic of Ireland: National University of Ireland Maynooth, 72-78.

Comber, A.J., Fisher, P.F. and Wadsworth, R.A., 2007. User-focused metadata for spatial data, geographical information and data quality assessments. *In*: M. Wachowicz and L. Bodum, eds. *10th AGILE International Conference on Geographic Information Science*, 8-11th May 2007. Aalborg, Denmark: Aalborg University, p.p. 13.

FGDC, 1998, FGDC-STD-001-1998. Content standard for digital geospatial metadata. Reston, VA, USA: Federal Geographic Data Committee, p.p. 90.

Halfawy, M., Vanier, D. and Froese, T., 2006. Standard data models for interoperability of municipal infrastructure asset management systems. *Canadian Journal of Civil Engineering*, 33,1459-1469.

Hart, D. and Phillips, H., 2001. *Metadata Primer - A "How To" Guide on Metadata Implementation* [online]. Available from: www.lic.wisc.edu/metadata/metaprim.htm [accessed 29 March 2009].

Hobona, G., James, P. and Fairbairn, D., 2004. Facilitating Data Discovery In Environmental Data Clearinghouses Through Spatial Data Mining *In*: A Lovett, ed. *Proceedings of the Geographical Information Science Research UK Conference (GISRUK) 12th Annual Conference*, 28-30th April 2004. Norwich, UK: University of East Anglia, 163-167.

ISO, 2005, BS EN ISO 19115:2005. Geographic Information - Metadata. Failand, Bristol, UK: BSi British Standards, p.p. 154.

Kim, T.J., 1999. Metadata for geo-spatial data sharing: A comparative analysis. *The Annals of Regional Science*, 33, 171-181.

Limbach, T., Krawczyk, A. and Surowiec, G., 2004. Metadata Lifecycle Management with GIS Context. In *10th EC GI & GIS Workshop, ESDI State of the Art,* 23-25th June 2004 Warsaw, Poland, p.p. 10.

Mathys, T., 2004. The Go-Geo! Portal Metadata Initiatives. *In*: A Lovett, ed. *Proceedings of the Geographical Information Science Research UK Conference (GISRUK) 12th Annual Conference*, 28-30th April 2004. Norwich, UK: University of East Anglia, 148-154.

Nebert, D.D., 2004. *Developing Spatial Data Infrastructures: The Spatial Data Infrastructure Cookbook v2.0* [online]. Available from: http://www.gsdi.org/gsdicookbookindex.asp [Accessed 29 March 2009].

Tsou, M.-H., 2002. An Operational Metadata Framework for Searching, Indexing, and Retrieving Distributed Geographic Information Services on the Internet. *In:* M. Egenhofer and D. Mark eds. *Geographic Information Science (GIScience 2002)*. Berlin, Germany: Springer-Verlag, 313-332.

A review of the geospatial metadata lifecycle from a public sector data clearinghouse perspective.

Research Paper I

Contributors: Bruce M. Gittings and Femke Reitsma

Abstract

Geospatial metadata play a key role in enabling the online discovery of geospatial data, their subsequent distribution and eventual reuse. Spatial data clearinghouses, whether as stand-alone metadata registers or as part of broader Spatial Data Infrastructures (SDI) have been predominant actors in facilitating the discovery of geospatial content by virtue of their surrogate metadata. Contribution of metadata to these clearinghouses is not always a seamless process however. The geospatial metadata lifecycle, from creation to publication, is characterised by a series of distinct steps that can complicate ongoing clearinghouse participation while potentially discouraging prospective donors. In the current paper we investigate a number of these challenges. With the UK's public sector GIgateway metadata service as a reference implementation, we explore opportunities for overcoming these challenges with a view to improving data accessibility. Proprietary and open source approaches are examined in the context of facilitating metadata publication, enhancing the service infrastructure as well as addressing a number of end-user considerations.

Introduction

The impact the World Wide Web (WWW) and other wide area networks have had on the sharing and transfer of information has been widely recognised (Yang *et al.* 2005, Rajabifard *et al.* 2006, Gahegan *et al.* 2007, Yang *et al.* 2007). Distribution has evolved from point-to-point exchange between predetermined actors to innumerable sources collectively publishing vast volumes of data that are widely accessible via the Internet. In light of such abundant and diverse resources, the merit of using metadata to discriminate between what may or may not be relevant is apparent (Greenberg *et al.* 2006).

In the Geographical Information Sciences (GIS) community, this merit has long been recognised (Nanson *et al.* 1995, Davey *et al.* 1996, Göbel *et al.* 1998, Craglia *et al.* 1999, Kim 1999, Lemmens *et al.* 2002, Mathys 2004, Schuurman *et al.* 2006). The spread of GIS software has fed the demand for geospatial data for a variety of applications, resulting in ever mounting volumes being produced (Guptill 1999, Deng 2002). The increasing likelihood of redundant data collection efforts, rising production costs and investment recovery motives all emphasize the importance of the location, evaluation and eventual exploitation of existing geospatial data holdings.

Geospatial data are particular however, characterised by geometry and encoded in ways that are impenetrable to the type of indexing techniques used by search engines for online documents (Batcheller 2008). They also tend to be unwieldy and instantiated in a variety of proprietary and open formats, making the direct and simultaneous exploration of multiple instances untenable over even the most efficient networks. The role geospatial metadata play as text-based data surrogates is therefore key in enabling the online discovery of geospatial datasets or *layers*, their subsequent distribution and eventual reuse.

Specialist systems designed to facilitate the above objectives have evolved since their emergence in the mid 1980s (Crompvoets *et al.* 2004). Spatial data clearinghouses, with their focussed model of metadata compilation and dissemination have been latterly augmented by geospatial portals and one-stops, combining data discovery and delivery within integrated online platforms. Spatial data infrastructures (SDI) incorporate such systems, encompassing not only the technology and data but also the standards, policy and participants therein. Whether first generation data-centric SDIs as described by Masser (1999) or those of subsequent generations that are increasingly process-based (Rajabifard *et al.* 2003), all have an underlying reliance on geospatial metadata as the means of accessing the data depicted.

Despite these advances, fundamental challenges remain when considering the metadata contribution lifecycle of geospatial data sharing initiatives. Whether relating to metadata generation, its maintenance and update, its exposure via discovery services or eventual use, technical impedances complicate ongoing participation while arguably discouraging prospective donors. In the current paper we detail a number of these challenges, and with the UK's public sector GIgateway metadata service as a reference implementation, investigate means of how they may be addressed. Further, as GIgateway offers a prime example of a first generation SDI founded upon the clearinghouse model, we take the opportunity to explore some of the technical limitations faced by such services and potential avenues for developing them.

The rest of the paper is structured as follows. We provide some background on the pervading climate that saw the emergence and development of the service now known as GIgateway. A description of the geospatial clearinghouse model follows, illustrating the principal actors in the metadata lifecycle and common bottlenecks therein. We propose a number of approaches designed to encourage the flow of metadata, and finally provide a perspective on how the contribution of metadata in facilitating data visualisation and access contributes to the evolution of pioneering clearinghouse architectures. A discussion on the issues raised during the course of the paper is followed by some final conclusions.

A brief history of GIgateway

Coordinated efforts in the UK to promote the exchange of geospatial holdings by way of metadata arguably began following the publication of the Chorley Report in 1987 and its call for 'data registers' (Heywood 1997). Prompting the rise of niche initiatives (e.g. the Intra-governmental Group on Geographic Information (IGGI)), it was not until the mid 1990s that a consortium led by the Ordnance Survey founded the UK-wide National Geospatial Data Framework (NGDF). The NGDF was initially conceived as a nebulous framework of applications and services, of standards and procedures to link existing data, to “formalise their availability, describe their quality… and provide tools for exploring the data” (Davey and Murray 1996). In spite of such high expectations, a disinterested government, the associated budgetary constraints, organisational reshuffles and an urgent desire for progress all provoked a rethink, and resulted in a more realistic first technical deliverable of a national metadata clearinghouse. Finally in 2000 the askGIraffe metadata service, forerunner of GIgateway, was deployed.

This stands in contrast to the situation in the United States where supporting legislation such as President Clinton’s Executive Order 12906 (1994) had been forthcoming, affording clear direction and prompting the foundation of the Federal Geographic Data Committee (FGDC). Indeed, where the FGDC’s National Geospatial Clearinghouse had been seen to flourish and its accomplishments informing such next generation services as the Geospatial One-Stop, the NGDF eventually folded. The askGIraffe service subsequently passed to a new custodian (the Association for Geographic Information (AGI)) where it was re-branded as GIgateway in 2003. Uncertainty nevertheless remained as to where the service’s ownership lay, its funding source and strategic goals, and led to its consolidation rather than expansion. For potential contributors, the benefits of involvement as a consequence were not readily apparent; ongoing participation meanwhile was arguably driven more by a desire to be seen to contribute or through some form of compulsion.

Fresh impetus came from European Union directives such as the reuse of Public Sector Information (PSI) in 2003 and the Infrastructure for Spatial Information in Europe (INSPIRE) in 2004. The latter in particular formalised requirements that member states facilitate location of and access to geospatial layers "for the purpose of formulation, implementation, monitoring and evaluation of Community policy-making"[10]. Expectations are that member states provide "discovery services" for geographic datasets specified in the directive using metadata conforming to the ISO 19100 standardisation series. Unsurprisingly, both directives have brought increasing attention to GIgateway as the UK's flagship service and how these expectations can be met.

The geospatial data clearinghouse model

A geospatial data clearinghouse provides a single point of access to multiple metadata repositories, distributed across the Internet and hosted on servers (nodes) maintained at each location. Each searchable repository holds standardised metadata that detail the content, condition and means of obtaining available data resources. The preceding could describe any metadata-based online catalogue; among the distinguishing functional characteristics of geospatial clearinghouses is the ability to search metadata by location. This is facilitated by the coordinates that frame each layer, captured in its metadata and used to identify those resources whose geographic coverage they spatially relate to, such as by overlapping, touching or enclosing.

[10] http://www.ec-gis.org/inspire/

First generation SDI largely implement geospatial data clearinghouses using a communication protocol based upon the GEO profile of the ANSI[11] information retrieval standard Z39.50-1995 (NGDF 1999). GEO, in detailing 'query-able' metadata elements, provides the schema against which all search and access operations are executed. In the case of GIgateway, data custodians submit metadata conforming to the UK GEMINI standard, a discovery profile based upon the ISO 19115 and e-GMS conventions. Records are subsequently transformed to the FGDC-derived GEO application profile during the metadata indexing process performed upon each node.

For the user, a typical metadata search is initiated using a web-based form on a central portal (Figure 1) using free text, keywords, dates or geographic coordinates or any combinations thereof. Query strings are generated and sent to participating clients where they are parsed and used to search each node's metadata index. Results are compiled and returned to the central gateway (portal) where they are collated for display in the user's browser where they are used to discern those layers of interest and how data custodians may be contacted.

[11] American National Standards Institute

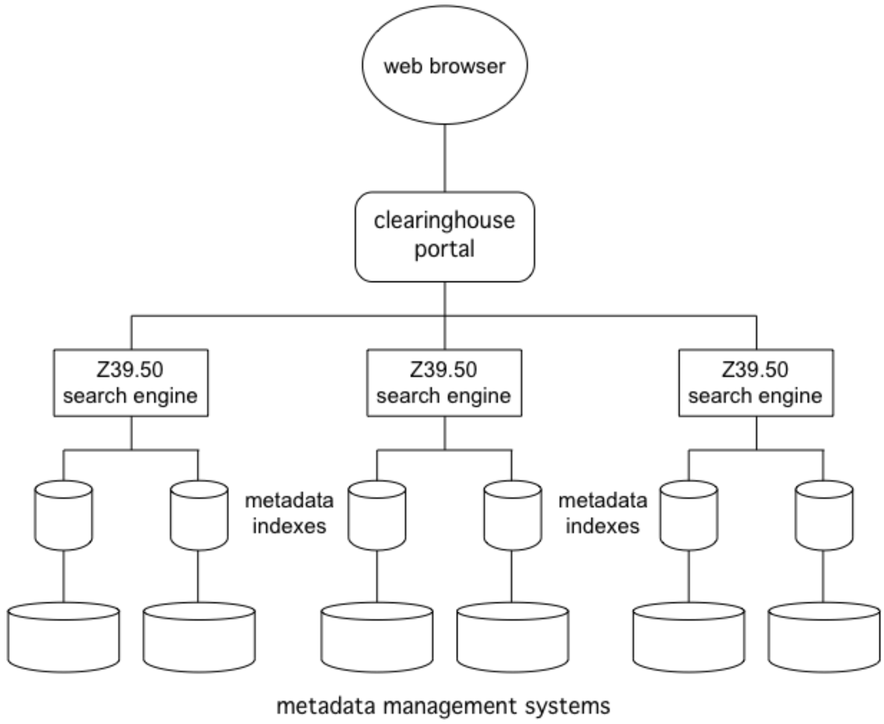


Figure 1. Geospatial data clearinghouse model

The metadata lifecycle

The continued provision and maintenance of metadata records sustain any geospatial data sharing initiative; their continued success therefore relies on those publishing to such initiatives. Existing contribution streams must be safeguarded and new ones encouraged; it is prudent therefore that perceived or actual barriers to participation are addressed. For clearinghouse-based services such as GIgateway, the path from metadata production to publication may be characterised by a series of distinct steps punctuated by repeated human intervention (Figure 2). Assuming data management controls are in place, metadata records are manually created or updated upon the creation or update of the dataset they depict. According to the needs of the contributing organisation, datasets may be documented internally by detailed[12] or

[12] Nebert (2004) discusses three levels of increasingly detailed metadata for geospatial data:

discovery metadata. As only discovery metadata are indexed and exposed, subsets of more expansive descriptions must be processed and deposited to the type of node employed.

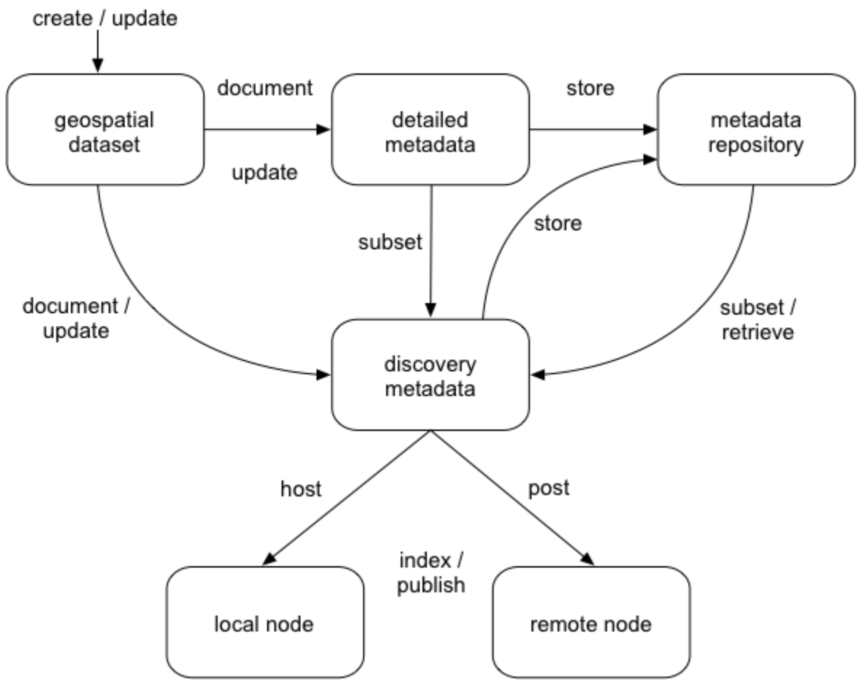


Figure 2. Typical geospatial clearinghouse metadata publication workflow. Arrows between workflow objects represent where manual input is necessary.

Metadata generation

Surmounting obstacles to the first phase of the metadata lifecycle – its creation – are crucial if service contribution streams are to be encouraged and maintained. Despite its apparent utility, authoring metadata can be perceived as a tedious, expensive or unnecessary drain on resources (West-Jr. *et al.* 2002, Mathys 2004). By simplifying the generation process, this perception may be countered and help mitigate its associated "human bottleneck" (Liddy *et al.* 2002).

---

discovery, exploration and exploitation. For current purposes, detailed encompasses the two latter levels.

Unlike the digital documents produced by conventional word processing software, geospatial datasets are of little or no immediate use in the absence of certain information that frame each instance. Details such as the feature type represented (e.g. point, line or polygon), coordinate projection, spatial resolution and geographic extent of the data must either be manually provided or consequently calculated by the GIS software in which the data are registered before meaningful analyses can be performed. In response to rising demand for documenting geospatial data assets, a number of GIS applications (e.g. ESRI's ArcGIS, Cadcorp's SIS) have evolved mechanisms for managing metadata documents in which these details are automatically recorded. Combined with the development platforms that typically accompany such software packages (e.g. Microsoft .Net, Sun Microsystem's Java), opportunities have consequently arisen for extending the base metadata management functionality for the purposes of computationally supporting metadata generation.

To illustrate this in the context of GIgateway, an existing approach is elaborated upon (Batcheller 2008). ESRI's ArcCatalog, the data and metadata management module of its popular ArcGIS suite, is customised to incorporate a UK GEMINI-specific authoring tool. Metadata elements are calculated using bespoke harvesting and extraction algorithms that draw upon three main sources – the dataset to be documented, its ambient computing environment and any pre-prepared content such as the default metadata previously described. Populating a form interface, elements are assessed for quality and blank entries completed prior to record output.

Individual records are exported to XML, validated against a UK GEMINI-compliant XML schema and ultimately transferred to a GIgateway node where they are exposed to the queries of service users.

Metadata management

Following record completion, the question of metadata management arises. To minimise the effort of contribution to services such as GIgateway, it is important to align the process of metadata provision with incumbent management practices. Considering the diversity of participants the service aims to cater for, this alignment is not trivial; protocols, resources and expertise will vary, as will the metadata storage techniques (Tyler 2002). Records are typically administered in one of three principal ways. One sees the metadata held alongside their associated data, facilitating a relative ease of update. Another is where the records are held in a detached relational database management system (RDBMS) so as to leverage secure and robust data management features (Date 2003). Finally, metadata may be stored as text-based files in a document repository for upkeep with basic text editors.

Further, records are rarely authored or updated in the same location in which they are exposed to discovery services. Multiple metadata instances, potentially encoded in more than one standard, must therefore be managed. Whether records are merely copied to where they are published or first require transformation to a conformant discovery profile as described above, update latency considerations arise. Similarly, concerns arise as to metadata record integrity and how they relate to not only the data depicted but also other metadata instances.

Migrating metadata storage in its entirety to a RDBMS presents one option for countering integrity concerns. With a database schema based upon the most detailed standard employed, multiple metadata instances may be merged into a single record, offering a solitary point of update. For instance, an organisation contributing to GIgateway stores exploitation metadata in a database repository; UK GEMINI fields form a subset of the more detailed standard schema but are not treated as distinct until called upon as discovery metadata elements. For enterprise GIS architectures based upon relational database back-ends there will be further benefits, as metadata will now be maintained in close proximity to their data and serve better integration of data editing and metadata update workflows, potentially within the same suite of software.

Database hosted metadata are exploited by discovery services in one of two ways. For clearinghouses based upon the Z39.50 protocol such as GIgateway, metadata records may be immediately accessible via a Z39.50-compliant database interface. No further metadata instances need be created, however this advantage is tempered by potentially restrictive performance consequences following the imposition of an extra software layer upon the service stack. An alternative method is the automated export of discovery metadata at defined intervals or upon specific database triggers such as record update, with XML files being output to Z39.50 index repositories for assimilation within the service.

For organisations that prefer not to adopt a database-driven approach, metadata integrity concerns and metadata 'silos' may be met through a system based around formal synchronisation (Figure 3). Developing the approach of Dunfey *et al*. (2006), a highly formalised procedure, a synchronisation file – which "acts as a road map for the system" – or synchronisation daemon provide the means of reconciling otherwise unconnected metadata instances. A metadata master copy is updated or created and synchronisation is initiated. Pre-existing metadata is updated or overwritten according to the storage strategy employed; new instances are imported or copied. Discovery metadata can be directly output to a clearinghouse node, whereas detailed metadata must first undergo transformation (subset).

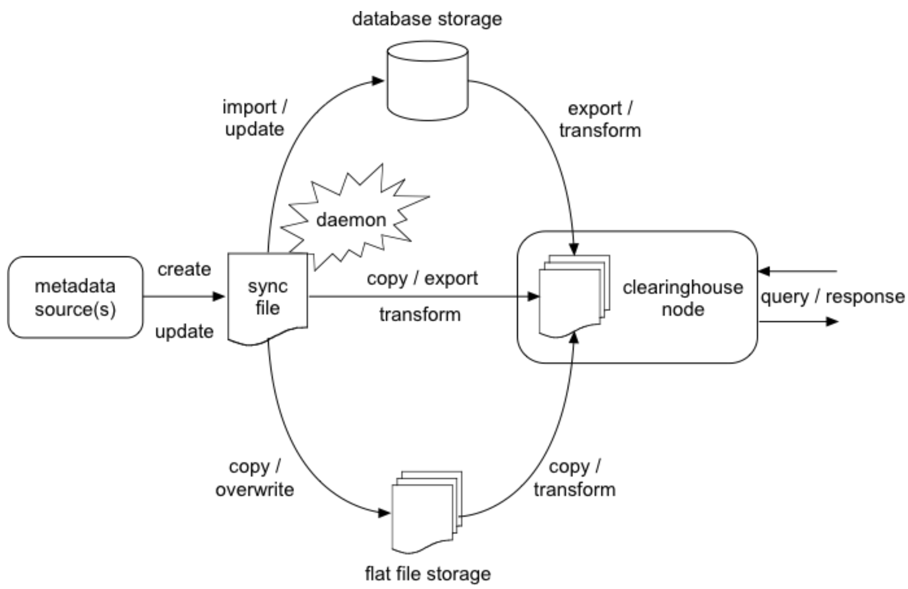


Figure 3. Formal metadata synchronisation

Metadata hosting

Clearinghouse contributors supply metadata either by transferring the records to an externally managed node or by exposing them on a node they themselves host. Organisations are typically encouraged to host their own node as this serves to foster a sense of ownership and participation amongst contributors (Nanson *et al.* 1995). Nevertheless, evidence suggests an unwillingness to do so due to certain “internal political problems and technical issues”[13]. While circumventing these political obstacles may well pose the greater challenge, options exist to tackle the technological concerns relating to node installation and maintenance.

Mounting a clearinghouse node involves the integration of a number of disparate components, including client-based metadata preparation, validation and parsing logic as well as the necessary metadata server software. The aforementioned unwillingness arguably stems from perceptions that the process requires a high-level of expertise resulting in setup being left to IT departments, outsourced to consultancies or indefinitely postponed where financial resources are insufficient. Associated hardware costs may also be prohibitive; bundling the necessary software components within an automated installation and configuration package on the other hand would serve to alleviate remaining concerns related to node configuration.

[13] AGI GIgateway Advisory Group Meeting Minutes, 18th May 2005: http://www.gigateway.org.uk/aboutus/aboutus.html

Metadata hosting by proxy

Clearinghouse participants opting not to host their own node may transfer metadata to nodes hosted elsewhere, such as is the case with GIgateway and its centrally managed metadata repository. Contribution may be by bulk transfer via physical media or online; services also frequently offer web-based metadata editing forms (e.g. GIgateway's MetaGenie[14]) for sequential record submission. Regardless of approach, resources are necessary to process each submitted record to ensure it is fit for publication.

Implementing a means by which pre-validated metadata may be automatically collected can remove much of these manual processing requirements. The Open Archives Initiative's Protocol for Metadata Harvesting (OAI-PMH) can provide one such option, functioning by the bulk retrieval of metadata records into a central location in a contrast to that employed by Z39.50 solutions. Conceptually it is viewed as substituting one node type (Z39.50) for another (OAI-PMH), but as there are few maintenance overheads aside from creating a web accessible folder, the approach mitigates some concerns associated with its management. Moreover, as the protocol is HTTP-based, no additional configuration and security measures are needed beyond those necessary for standard web servers (Amin 2003). Used in conjunction with the metadata generation and validation tool above, contributors can deposit quality-controlled records into the OAI-PMH folder from where they may be

[14] http://www.gigateway.org.uk/metadata/metagenie.html

harvested and consequently ingested into the clearinghouse service – without the need to open a dialogue between supplier and host.

Metadata maintenance

The role of contributor should not end with metadata submission, but persist so that records are maintained to accurately reflect the data they describe. A date stamp indicating when a record was last changed is commonly implemented in metadata standards to capture evidence of this maintenance; UK GEMINI details a *Date of update of metadata* element. Assessing whether to pursue data using this field is not without complications, as static or infrequently updated datasets may be incorrectly perceived as outmoded where an accurate yet long passed date is evident; forming a new element for registering the date of metadata review would serve to improve decision support in these instances.

There is of course no guarantee that metadata records will be maintained once published, regardless of what the element set has been designed to capture. A formally enforced obligation to maintain up to date surrogates may function in instances where financial, political or other leverage exists, but whether such an approach is sustainable must be open to question. As a possible alternative, a mechanism for automated notification is combined with a means for grading the maintenance history of a metadata instance. Metadata custodians can be contacted at regular intervals based upon a given timestamp and email address encoded in the record, reminding them to perform a review and update, if necessary. Maintenance history grades will allow records to be evaluated at a glance for staleness by users while compelling contributors to regularly maintain records lest their holdings be unfavourably compared with those of the wider community.

Beyond first generation clearinghouses

In the overall context of SDIs, geospatial metadata are pivotal but remain ultimately a means to an end. Nevertheless, clearinghouse users are confronted by certain impediments towards achieving this end, such as an uncertainty that the metadata unambiguously reflect the data sought, and no immediate, clearly defined route to securing the desired data once identified. While such considerations may contribute to replacing or supplementing existing initiatives (e.g. the National Geospatial Clearinghouse and the complementary Geospatial One-Stop in the US), mounting ones from scratch is not always feasible. On the other hand, persisting with an outdated system when faced with rising user expectations is rarely advisable (Fulker 2003, Rackham 2004), particularly when its success is invariably linked to the degree to which user requirements are met (Masser 1999).

Geographic web services complying with Open Geospatial Consortium (OGC) standards offer options for incorporating data exploration and delivery without the need for wholesale revision of existing clearinghouse architectures. They further enable datasets to be rendered according to their terms of availability. Web Map Services (WMS) can be used with licensed layers as they render static representations of data, not the data themselves. Datasets with no associated acquisition cost may be served using Web Feature Services (WFS) and Web Coverage Services (WCS), for vector data and imagery respectively; both formats render the actual, fully accessible data and provide for layer visualisation coincident with delivery.

Such OGC web services render their underlying data, or representations thereof, within standard web-based browsers. They are invoked via Uniform Resource Locators (URLs), and thus provide a way of linking metadata to the geospatial data they depict. In the case of GIgateway, once this URL is generated for a given layer it can be used to populate the UK GEMINI *Browse Graphic* field and disseminated as part of the dataset's metadata record.

Providing for access to licensed data following visualisation requires the inclusion of an online transaction model, whether catering for direct purchase, service-level licensing or a combination of the two. While the intricacies of such models are outside the scope of the current work, Figure 4 provides a conceptual overview of how an Internet Payment Service Provider (IPSP) solution could be incorporated within a W*S-extended clearinghouse architecture.

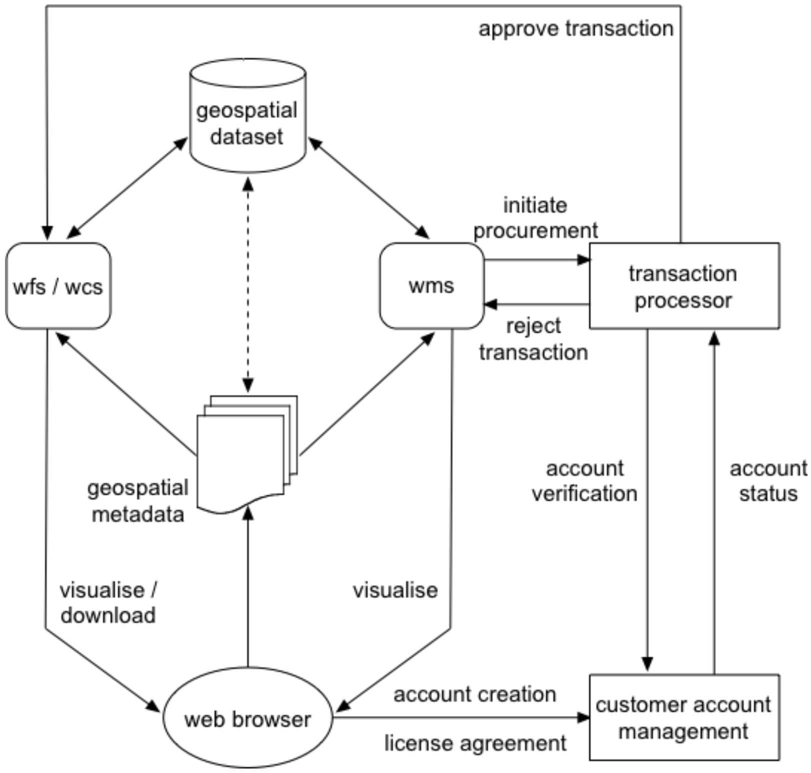


Figure 4. Incorporating data visualisation, access and transaction support within the base clearinghouse model

Supporting system architecture

While provided in the context of standard first generation clearinghouses, none of the proposed solutions elaborated herein are necessarily bound to their service infrastructures; the reality is that all geospatial data sharing initiatives based on metadata will face similar challenges. With this in mind, we move on to contemplate a more extensive overhaul of clearinghouse-based SDI. We described in the previous section how to extend basic clearinghouse functionality in a manner that would, by the criteria given by Crompvoets *et al*. (2004), facilitate transitioning an SDI from first to second generation. All the same, building upon legacy Z39.50 systems originally developed in the library community and that have been essentially retrofitted for application in the geospatial domain with little scope for interoperability going forward offers little by way of future proofing service infrastructures. Pursuing such a strategy does nevertheless allow for a quick, short-term fix as well as provide an opportunity for initiatives without the resources for a more involved overhaul.

Flexible and efficient for near transparent querying of multiple metadata repositories, Z39.50-based solutions have been perceived to be functionally limited in a number of ways. Troll and Moen (2001) for example question the ongoing utility of Z39.50 given its complexity and interoperability handicaps while Tsou (2002) comments that the inability to impose groupings upon items in potentially voluminous result sets can prevent identification of otherwise suitable datasets. In the geospatial domain, the variation in how the abstract protocol is instantiated results in inconsistent support for spatial searching, while its ability to scale is also called into question (Medyckyj-Scott *et al.* 2001, Amin 2003). As for the underlying GEO profile, the metadata transformation it necessitates imposes an added inefficiency to the metadata lifecycle, adding to the view that it remains inflexible in the face of evolving requirements (AGI 2007). Rocha and Henriques (2004) meanwhile argue that the changing face of geographical information services, with increased demand for mobile solutions, real-time, data-ready applications and the long-term aim of data retrieval in the absence of human mediation dictates the adoption of a different paradigm.

The emerging OGC Catalogue Service Specification 2.0.2 (OGC 2007) aims to provide for such a different paradigm. Adhering to the trend in which the development of geographical information technologies continue to be more closely aligned with the mainstream IT industry and interoperability efforts (Higgins *et al.* 2005), the specification details "an open, standard interface that enables diverse but conformant applications to perform discovery, browse and query operations against distributed and potentially heterogeneous catalog servers" [15]. Defining a number of communication protocols (bindings) based on CORBA, HTTP (or Catalogue Services for the Web, CSW) and a new iteration of Z39.50, adherence to the Specification enables creation of custom applications through the use of application profiles. Interoperability between different bindings is intended through the use of a minimal abstract OGC_Common Catalogue Query Language, providing further support for spatial query constructs including DISJOINT, INTERSECT, WITHIN and OVERLAP (OGC 2007).

With an existing clearinghouse such as GIgateway in mind, complications quickly emerge when considering the practical implications of transitioning its core infrastructure. Despite outlining a more sophisticated, yet open, treatment for geospatial resource discovery, the Catalogue Service Specification remains an abstract document, with few well-tested or mature implementations – hence the persistence of legacy Z39.50 solutions. Indeed, as Craglia *et al.* (2007) point out many aspects of the specification are ambiguously defined; different interpretations

[15] OGC Press release http://www.opengeospatial.org/press/?page=pressrelease&prid=188

of the specification resulting in non-interoperable bindings in spite of the best of intentions. If a technological transition is to occur, the ability to adapt structural components with minimal impact to service provision is vital.

To this end, OGC's Geospatial Portal Reference Architecture ((OGC 2004), Figure 5) presents a promising basis for developing a technological backbone of an SDI. The architecture allows for a set of vendor-agnostic interfaces that allow a core system to be put in place, modularly augmented or exchanged. *GeoNetwork opensource* for instance, a collaborative development effort of the FAO, UNEP and WFP, offers a freely available, partial implementation of the architecture. Its portal and catalogue components may be used on their own or combined with both commercial and open source solutions for data visualisation (Portrayal Services) and data access or processing (Data Services). Additionally, available bindings include Z39.50 and thus offer the potential for incorporation within or replacement of incumbent clearinghouse architectures. Whilst not offering a departure from the protocol as espoused by Tsou (2002) or Rocha and Henriques (2004), its open, modular architecture provides scope to replace the communication protocol as laid out in the OGC's Catalogue Service Specification, as well as facilitating interoperability with emerging national and international schemes, as propounded by the INSPIRE directive.

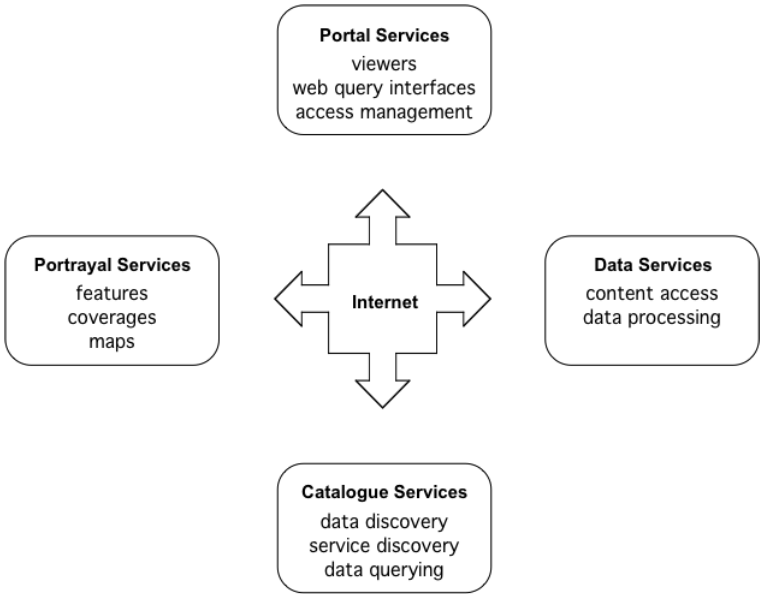


Figure 5. Overview of OGC Portal Reference Architecture

Initial Recommendations

Prospective efforts to reinvigorate any clearinghouse will predictably be fraught with difficulty. Aside from future visions of how a service will be manifested, questions as to the prudence of jeopardising a potentially long history of investment in the incumbent technology, infrastructure and expertise certainly arise. In the case of GIgateway, the track record of its technical core is proven within the UK context and underpins what remains a popular and dependable, if limited, service. Considering the diversity of its stakeholders and the resistance to change witnessed in some quarters, strong reasoning for any overhaul will be required. Even in the event of a consensus, damage to the reputation of the given initiative could prove fatal if an enhanced service proves unreliable or does not live up to user expectations. Of course, initiating changes in service paradigm and maintaining the service thereafter will depend on whether financial and human resources are forthcoming.

Some of the development paths elaborated above raise further issues. With respect to coupling automated metadata generation with dataset editing workflows, the lack of open, standard geo-interfaces or Application Programming Interfaces (APIs) across the GIS industry currently precludes the creation of a universal solution, thereby necessitating the development of package-specific strategies. Automating metadata management and submission processes will serve to reduce the resources necessary for contribution to clearinghouse-based SDI, but do underline the need for quality and validation safeguards to ensure that inappropriate records are not disseminated. Any implementation of the solutions suggested above should be buttressed by systematic human-mediated quality control, preferably performed by appropriately trained users, whether on a spot-check basis or 'brute force' evaluation of all applicable metadata records.

For organisations with few records, or datasets that change infrequently, manually generating, updating and submitting records may well represent the preferred way forward. Similarly, preference for retaining some manual control over automated processes should not be discounted, particularly for those already with well-defined protocols in place or those reluctant to yield control to what may be perceived as a 'black box' procedure. In any case, focus should remain on promoting quality metadata contribution, not the excessive imposition of further layers of complexity on the process where it is not wanted nor warranted.

DBMS techniques would by their nature provide for better management and integrity of geospatial metadata. While there is an argument that suggests this would significantly add to the complexity of the system, the well-established interfaces to DBMS based on SQL (Structured Query Language) should render such components appropriately modular. Although issues of cost may be raised as concerns, free and open source software (FOSS) such as MySQL and Postgres are viable options.

Both proprietary and FOSS solutions have been discussed – each has its place, with their own particular advantages and disadvantages. Proprietary systems can be argued to offer stability, less risk and provide buy-in to a ready-made, presumably well-tested product complete with support. Yet they can prove expensive. FOSS can provide a less expensive alternative, although are rarely completely free, often requiring specialised expertise whether in-house or out-sourced. What is crucial is to ensure the modularity of components linked by standardised interfaces such that there should be no dependence on either proprietary or FOSS because these components can be readily replaced.

Additional flexibility can be conferred by providing the aforementioned software complete with their source code, whether crafted in proprietary or open environments. While universally applicable solutions are presented, enabling access to the inner workings of such software will ease integration efforts with incumbent configurations that invariably differ between organisations. Moreover, by providing support for facilities similar to those of the online open source communities (e.g.

SourceForge), namely a code repository and a user forum, enthusiastic participants can further develop, discuss and distribute provided solutions in a collaborative setting to the benefit of the wider participating community.

Providing robust account management, secure access and scalable computing resources are critical, particularly if offering facilities for data visualisation and download are successful in attracting more users to the service. Usage should therefore be closely monitored to enable a proactive response to potential increases in traffic volumes. And with the proposed provision of both free and licensed data, such monitoring systems could be extended to grant data procurement analyses and feedback mechanisms, as well as offer a potential test-bed in which the implications of supplying free versus licensed data can be analysed.

The INSPIRE and PSI directives alone will most likely be insufficient in affording the momentum necessary to drive the geospatial metadata generation and contribution necessary for tapping underexploited geospatial resources. Marketing campaigns and educational drives (such as workshops and seminars) can continue to complement and reinforce other facets of data sharing initiatives, help raise awareness of the benefits of, and lower barriers to, participation.

Future Prospects

A range of technical options for enhancing the flow and effectiveness of geospatial metadata has been presented in the context of legacy clearinghouses and a reference implementation. As components of SDI, it is worth reinforcing the point that clearinghouses remain facilitators of the wider aim to allow the discovery, exchange and utilisation of geospatial data. Clearinghouses of the first generation provide a single point of access to distributed metadata repositories, but these metadata simply signpost where data are found, not facilitate immediate data access. Second-generation systems, characterised in part by the introduction of complementing web services, typically facilitate access to and manipulation of geospatial data. For initiatives relying upon legacy technology, transitioning from one generation to the next may be performed stepwise or in wholesale fashion; whichever path undertaken, sustaining the supply, maintenance and flow of metadata should remain important concerns. Providing a means of accessing data and forging linkages to other similar schemes at national and international level should only be contemplated once such concerns have been satisfied.

In the specific case of GIgateway and the diverse nature of its stakeholders, any decision on how to evolve the service will never be based purely on the technological. Indeed, ambiguities surrounding its purpose, technological expectations, funding sources, strategic direction and especially where its ownership ultimately lies must be resolved before it can move forward. Interoperability with other services (not to mention compatibility with the UK GI Strategy) must remain a critical factor, particularly in light of legislative requirements at national and European level. The need to maintain the service's standing in the face of emerging initiatives more in tune with both contributor and consumer expectations is also crucial to avoid perceptions of complacence and the resulting implications on numbers contributing to and exploiting GIgateway. Whatever path is ultimately taken, the overall objective should not only be the realisation of a service befitting that of an internationally visible initiative, but one that its users view as being fit for purpose.

References

AGI, 2007. Future Technology for a UK Metadata Service: Research Report. London, UK: AGI, p.p. 41.

Amin, S., 2003. The Open Archives Initiative Protocol for Metadata Harvesting: An Introduction. *In: DRTC Workshop on Digital Libraries: Theory and Practice*, 10-21st March 2003. Bangalore, India: DRTC, p.p. 13.

Batcheller, J. K., 2008. Automating geospatial metadata generation—An integrated data management and documentation approach. *Computers and Geosciences*, 34 (4), 387-398.

Clinton, W., 1994. Coordinating geographic data acquisition and access: the national spatial data infrastructure. Executive Order 12906, Federal Register 59. Washington D.C., USA: USG, 17671-17674.

Craglia, M., Annoni, A. and Masser, I., 1999. Geographic information policies in Europe: national and regional perspectives. In *Proceedings of the EUROGI-EC Data Policy Workshop, Report EN19522 EN* 15 November 1999 Amersfoort, Netherlands. Ispra, Italy: Joint Research Centre, p.p. 28.

Craglia, M., Kanellopoulos, I. and Smits, P., 2007. Metadata: where we are now, and where we should be going. *In*: M. Wachowicz and L. Bodum, eds. *10th AGILE*

International Conference on Geographic Information Science, 8-11th May 2007. Aalborg, Denmark: Aalborg University, p.p. 7.

Crompvoets, J., Bregt, A., Rajabifard, A. and Williamson, I., 2004. Assessing the worldwide developments of national spatial data clearinghouses. *International Journal of Geographical Information Science,* 18 (7), 665-689.

Date, C. J., 2003. *An Introduction to Database Systems, Eighth Edition*. Boston, USA: Addison Wesley.

Davey, A. and Murray, K., 1996. Update on the National Geospatial Database - Collaboration between Organisations. *In Proceedings of the 8th Conference of the Association for Geographic Information at GIS96*, 24-26 September 1996. Birmingham,UK. London, UK: AGI, 1-6.

Deng, Y., 2002. *The Metadata Architecture for Data Management in Web-based Choropleth Maps* [online]. Available from:
http://www.cs.umd.edu/projects/hcil/census/JavaProto/metadata.pdf
[accessed 29 March 2009].

Dunfey, R. I., Gittings, B. M. and Batcheller, J. K., 2006. Towards an Open Architecture Vector GIS. *Computers and GeoSciences,* 32 (10), 1720-1732.

Fulker, D., 2003. Metadata Strategies to Address NSDL Objectives. *In Proceedings of the 5th Russian Conference on Digital Libraries RCDL2003*, St-Petersburg, 29-31 October. St. Petersburg, Russia: St. Petersburg University, p.p. 4.

Gahegan, M., Agrawal, R., Banchuen, T. and DiBiase, D., 2007. Building rich, semantic descriptions of learning activities to facilitate reuse in digital libraries. *International Journal on Digital Libraries,* 7, 81-97.

Greenberg, J., Spurgin, K. and Crystal, A., 2006. Functionalities for automatic metadata generation applications: a survey of metadata experts' opinions. *International Journal of Metadata, Semantics and Ontologies,* 1 (1), 3-20.

Göbel, S. and Lutze, K., 1998. Development of meta databases for geospatial data in the WWW. *In 6th International Symposium on Advances in Geographic Information Systems, ACM-GIS '98*, 6-7 November 1998 Washington D.C., USA. New York, USA: ACM, 94-99.

Guptill, S. G., 1999. Metadata and data catalogues. *In:* P. Longley, M. F. Goodchild, D. J. Maguire, and D. W. Rhind, eds. *Geographical Information Systems*. Chichester, UK: Wiley, 677-692.

Heywood, I., 1997. *Beyond Chorley - current geographic information issues*. London, UK: AGI.

Higgins, C., Robertson, A. and McGarva, G., 2005. Edinburgh University Data Library Geographic Information Standards Final Report. Edinburgh, UK: EDINA p.p. 37.

Kim, T.J., 1999. Metadata for geo-spatial data sharing: A comparative analysis. *The Annals of Regional Science*, 33, 171-181.

Lemmens, R. and de By, R.A., 2002. Distributed GIS and metadata - Methods for the description of interoperable GIS components. *In International Workshop on Mobile and Internet GIS,* August 15-16 2002 Wuhan, China, p.p. 9.

Liddy, E., Allen, E., Harwell, S., Corieri, S., Yilmazel, O., Ozgencil, N. E., Diekema, A., McCracken, N. J., Silverstein, J. and Sutton, S. A., 2002. Automatic metadata generation and evaluation. *In Proceedings of the 25th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, 18-21 November, Tampere, Finland. New York, USA: ACM, 401-402.

Masser, I., 1999. All shapes and sizes: the first generation of national spatial data infrastructures. *International Journal of Geographical Information Science,* 13 (1), 67-84.

Mathys, T., 2004. The Go-Geo! Portal Metadata Initiatives. *In*: A Lovett, ed. *Proceedings of the Geographical Information Science Research UK Conference*

(GISRUK) 12th Annual Conference, 28-30th April 2004. Norwich, UK: University of East Anglia, 148-154.

Medyckyj-Scott, D., Chappell, C., Pradhan, A. and O'Hanlon, C., 2001. A geo-spatial data resource discovery tool for UK Further and Higher Education - Project Overview and Recommendations. Edinburgh, UK: EDINA, p.p. 60.

Nanson, B., Smith, N. and Davey, A., 1995. What is the British National Geospatial Database? *In 7th Conference of the Association for Geographic Information*, 22 November 1995, Birmingham, UK. London, UK: AGI, 1-8.

Nebert, D.D., 2004. *Developing Spatial Data Infrastructures: The Spatial Data Infrastructure Cookbook v2.0* [online]. Available from: http://www.gsdi.org/gsdicookbookindex.asp [Accessed 29 March 2009].

NGDF, 1999. Communication Protocols for a Distributed Geospatial Metadata Service. London, UK: AGI, p.p. 15.

OGC, 2004. Geospatial portal reference architecture: a community guide to implementing standards-based geospatial portals. Wayland, MA: OGC, p.p. 17.

OGC, 2007. OGC Catalogue Services Specification 2.0.2. Wayland, MA: OGC, p.p. 204.

Rackham, L., 2004. An Independent Review of the Sustainability of a UK Metadata Service for Geographically Related Information. London, UK: AGI / OS, p.p. 49.

Rajabifard, A., Binns, A., Masser, I. and Williamson, I., 2006. The role of sub-national government and the private sector in future spatial data infrastructures. *International Journal of Geographic Information Science,* 20 (7), 727-741.

Rajabifard, A., Feeney, M. E., Williamson, I. and Masser, I., 2003. National SDI-initiatives. *In:* I. P. Williamson, A. Rajabifard, M. F. Feeney eds. *Developing Spatial Data Infrastructures: From Concept to Reality.* London, UK: Taylor & Francis, 95-109.

Rocha, J. G. and Henriques, P. R., 2004. Towards XML Web Services based Clearinghouses. *In 7th Global Spatial Data Infrastructure Conference*, 2-6 February 2004. Bangalore, India, p.p. 14.

Schuurman, N. and Leszczynski, A., 2006. Ontology-based metadata. *Transactions in GIS*, 10 (5), 709-726.

Troll, D. and Moen, B., 2001. Report to the DLF on the Z39.50 Implementers' Group - Moving Towards the Future of Z39.50. Issues and Options Based on ZIG Meeting Discussions December 6-7, 2000 [online]. Available from:
http://www.diglib.org/architectures/zig0012.htm
[Accessed 29 March 2009]

Tsou, M.-H., 2002. An Operational Metadata Framework for Searching, Indexing, and Retrieving Distributed Geographic Information Services on the Internet. In M. Egenhofer and D. Mark eds. *Geographic Information Science (GIScience 2002), Lecture Notes in Computer Science Vol. 2478*. Berlin, Germany: Springer-Verlag, 313-332.

Tyler, G. J., 2002. *Managing Metadata: Developing technical solutions for the askGIraffe geospatial metadata gateway*. Thesis (MSc). University of Edinburgh, UK.

West-Jr., L.A. and Hess, T.J., 2002. Metadata as a knowledge management tool: supporting intelligent agent and end user access to spatial data. *Decision Support Systems*, 32, 247-264.

Yang, B., Purves, R. and Weibel, R., 2007. Efficient transmission of vector data over the Internet. *International Journal of Geographical Information Science,* 21 (2), 215-237.

Yang, C. P., Wong, D. W., Yang, R., Kafatos, M. and Li, Q., 2005. Performance-improving techniques in web-based GIS. *International Journal of Geographical Information Science,* 19 (3), 319-342.

A Geographical Information Projects Registry for Scotland

Research Paper II – short article

Contributor: Bruce M. Gittings

Introduction

Geographical Information (GI) systems, services and related data have long since proliferated beyond the confines of the specialist domains in which they were first devised (Tait 2005, Bédard *et al.* 2007). Public, private and academic organisations not traditionally versed in geospatial practices have become increasingly reliant upon GI disciplines to help deliver products, provide services and drive research (Goodchild *et al.* 2004, Harvey *et al.* 2006). What resulted not only saw an explosion in the volumes of data produced (Movva *et al.* 2005, Bertolotto *et al.* 2006), but also the associated tools, applications and accompanying expertise, spread across a range of disciplines (Annoni *et al.* 2003, Crompvoets *et al.* 2004, Bilasco *et al.* 2007).

Partnerships between private, local and central government agencies and research bodies have been key in driving such developments. This wider involvement nevertheless comes with its own implications, especially when considering the diversity of prospective collaborators. Disparities in data format, content vocabularies and information exchange protocols necessitate the use of collating techniques at destination (Devogele *et al.* 1998). This costs time and resources, and yet may still succumb to the loss of detail in the data being transferred, its misinterpretation or misapplication (Devillers *et al.* 2005, Bédard *et al.* 2007). Identifying and contacting appropriate domain specialists for expert counsel meanwhile is time-consuming at best where no tangible avenues of communication already exist. Equally, few tasks are truly unique, but identifying appropriate partners who may have achieved portions of one's own goals can be difficult. Collaborative efforts therefore offer potential savings in time, effort and resources, and are most effective when underpinned by mechanisms and protocols to manage and exchange data, expertise and other related geographic information in a coherent, integrated and effective manner.

"One Scotland – One Geography"

It was in recognition of these issues that in 2005 the Scottish Executive[16], supported by the Association for Geographic Information (Scotland), launched the GI Strategy for Scotland titled "One Scotland – One Geography" (Easton 2005). Providing a framework in which detailed policies can be developed, the Strategy sets out to promote as wide use of GI as possible, supporting the faces, places and spaces of Scotland.

Spaces – a representation of a physical geographic object that can be defined on a map, such as rivers, streets or administrative areas and boundaries
Places – address information and postcodes along with textual information on place names
Faces – information on the people, either past, present or projected, within a certain area

[16] Now the Scottish Government, renamed in 2007

Endorsed by a broad range of stakeholders from the public, private and academic sectors, the Strategy outlines a five-year roadmap for action commencing in 2006, intending to promote efficient government through the re-use of ideas, data and systems. Drawn up by the Executive, specific goals include:

- provision of strategic vision and leadership to ensure an inclusive, co-ordinated approach to GI in Scotland
- delivery of accurate, up-to-date GI
- development of avenues through which GI can be shared
- promotion of the benefits of GI across all sectors
- promotion of the appropriate technical and professional standards for the efficient and effective use of GI in Scotland

While the initial focus is on central government and infrastructural development, the need for continued involvement of the wider GI community remains a central tenet for continued success. For such involvement to be effective however, opportunities for improved engagement between community members must be identified and promoted. Wider visibility of GI-related activity across Scotland is necessary, not only to avoid duplication of effort and misdirection of resources, but to facilitate knowledge transfer, the cross-fertilisation of ideas and to provide a showcase for such activities.

The UK's national geospatial data exchange service

At the national level, GIgateway serves as the flagship public sector metadata service for the United Kingdom. Arising from a number of predecessors, most notably the National Geospatial Data Framework (NGDF) and *askGIraffe*, GIgatway's *raisons d'être* remain that of its forerunners: "to increase the use of geospatial data; to facilitate development of markets for data and services"; and to "'future proof' investments and enhance decision-making through use of better information" (Rhind 1997, GIgateway 2003). The service works towards these objectives through the support of a distributed web-based network focussed on serving geospatial metadata – layer-level metadata designed to provide a means of identifying where the datasets they describe are held, and hence, exchanged. Users submit metadata queries via a central gateway, essentially a web-based form that accepts keywords and geographic extents as search parameters. Processed queries are sent to remote distributed clients that expose the metadata of participating organisations, and return search results to the gateway, where they are displayed.

Despite being a successful and well-respected clearinghouse for facilitating geospatial data exchange, GIgateway on its own was deemed insufficient to meet all of the needs of the GI Strategy for Scotland. Most notably, GIgateway remains a layer-level metadata service, to the exclusion of documentation for other GI granularities. As a consequence, they convey little of the organisational or socio-political context in which datasets are created and curated (Comber *et al.* 2007). Differing perspectives can sway how geography is modelled for a given area, particularly where there are contending motives behind data collection and collation.

What spurs data creation, who funded it, and other factors that influence the processes involved can for instance provide further criteria with which to evaluate GI initiatives and the data they may produce.

Further, due to the data layer-centric nature of GIgateway, successful use of the service can demand specialist knowledge as to what the clearinghouse has available, a particular problem for the uninitiated. Familiarity with the service, such as the data-specific terminology employed and what terms prove most effective in driving searches must be developed. Providing information at a higher level of abstraction via project documentation can assist with subsequent layer-level searches and offer a point of entry for initial exploratory investigations. As for expert counsel, whether on specific data layers or the techniques used to generate them, services like GIgateway offer only a route to the data custodian, not the subject matter experts who are likely to be most familiar with the circumstances in which the data are gathered.

The GI Project Registry

It was with these issues in mind that the Edinburgh Earth Observatory – the locus for GI and Remote Sensing research at the University of Edinburgh – in collaboration with the Scottish Executive and AGI-Scotland, developed an online directory providing details of GI-related projects, implementations and research activities. The 'GI Project Registry' (www.gisprojects.net), through the use of project documentation, aims to help identify and promote synergies in step with the GI Strategy for Scotland's overall objectives outlined above. Essentially a project-level metadata approach, the Registry (also known by the acronym GIPR) can be used to provide insight into the wider context in which data are generated and used, with the implication of better informing data gathering operations.

Providing a means in which initiatives may be recorded and searched, participants are not only able to publicise their own activities, but also gain an insight into what is going on elsewhere in the wider community. By offering such a virtual 'shop window' it is intended to encourage dissemination of expertise, identify opportunities for collaboration and forge lasting linkages across all GI related sectors. As a user-driven service, it is intended that the site be self-sustaining, with responsibility falling on each participant to maintain and update their own records.

Site Architecture

One of the principal specifications of the Registry was that it be quick to develop and easy to maintain to ensure that the service retained an agility to respond to evolving user expectations. Further, due to initial funding constraints, a solution built from open source components was the most desirable approach. A software stack based upon the LAMP model was accordingly configured for deployment: *Linux* provides the operating environment underpinning the service; *Apache* acts as the web server to accept user requests and transmit Hypertext Mark-up Language (HTML) documents to client browsers; *MySQL* serves as the underlying relational database system to store all the data and *PHP*[17] operates as the scripting language through which the

[17] PHP is the recursive acronym for PHP: Hypertext Preprocessor

components are inter-connected. The system design of the GIPR stands in contrast to the distributed architecture of GIgateway. The GIPR software stack is consolidated in one location, with a single point of contribution for project documentation.

User contributions

Development of the Registry proceeded in tandem with a study performed by the Scottish Executive seeking to identify stakeholders willing to participate. Once identified, each stakeholder was contacted in turn, informed as to the purpose of the Registry, and asked for information on any GI-related activity they deemed appropriate for inclusion. The definition of a 'GI-related activity' was deliberately open and non-prescriptive to encourage as broad an involvement as possible, although it was made clear that dataset-specific information should be contributed to GIgateway or a similar geospatial metadata service, such as the academia-focussed *Go-Geo!* initiative (Mathys 2004). Data collected during the study was used to populate the website prior to launch not only to provide a service with immediate value but to encourage further contributions from users presumed more likely to participate on discovery of a fully-functional solution.

Accounts were assigned to stakeholders following the data collection phase; subsequent users of the site willing to partake are prompted to register prior to contributing their own initiatives. Once registered, users gain access to the site's Members Area, enabling management of account specific information. Here, contributors can update their account details, add and edit their own initiatives as well as review contributions made.

It was recognised that the number of fields used to describe an initiative should be kept to a minimum to encourage contribution and to prevent it from being cumbersome. Nevertheless, the proposed fields were agreed with the stakeholders to be sufficiently comprehensive and meaningful. Fields marked as mandatory were similarly kept to a minimum, although users are encouraged to provide as many details as is possible. In practise, the standard used was a Qualified Dublin Core[18] profile of twelve elements as shown in Table 1. At present, only English language projects are catered for, although this may be reconsidered later. As for the rights and format elements, as the service does not deal with data per se, these were deemed unnecessary.

Dublin Core element	GIPR element / compound element
Creator	Record custodian
Title	Title
Subject	Keywords
Description	Abstract
Publisher	Record custodian
Contributor	Contact information
Date	Timespan
Type	Initiative category
Format	-
Identifier	Record ID; Initiative code
Source	Contact information; Record custodian
Language	-
Relation	URL of GIPR
Coverage	Geographic area; Extent
Rights	-

Table 1. The Qualified Dublin Core profile used in the GIPR and the corresponding 'user-friendly' fieldnames used in the web interface

[18] http://dublincore.org/

Entries provide the basic details of each initiative, its geographical area of coverage, its projected or realised time span and points of contact. Recorded details are managed by either a contributing user or assigned delegate, either of whom effectively become the custodian for the record in question.

Identifying Linkages

Use of the site is not limited to those registered – anonymous users may browse and search recorded initiatives at will. Whatever the case, it is using the power of the site's search facility that linkages between initiatives can be identified. Free text search may be performed on the Title, Abstract or Keyword fields, or limited to specific fields within the database. Specific fields such as *Initiative Category* or *Sponsoring Organisation* for instance may be used in queries to locate initiatives within similar domains; queries on *Area of Coverage* and *Local Authority* will uncover those occupying the same geographic space, whereas those on *Initiative Start Year* and *Projected Year of Completion* will pinpoint activities that are currently active or of historical interest. Contact details for each record are provided, offering an appropriate point of contact should further information be required and facilitating internetworking between participants without the need for an intermediary.

A First Step

The 'GI Project Registry' is seen as an initial yet important step in underpinning the GI Strategy for Scotland, with emphasis on the resources critical to the Strategy's success – people, their expertise and how they may be linked to provide an invaluable resource. Detailing over two hundred initiatives shortly following its inception in Spring 2006, it has successfully raised awareness of and provided documentation on GI-based projects across Scotland. In order to maintain the value of the GIPR however, options for further developing the site, accounting for evolving user expectations and aimed at preserving their engagement, should be regularly monitored and implemented when possible.

Highlighting similar initiatives to those being viewed by the user can proactively expose synergies between existing projects, rather than leaving such discoveries to search result sets. An automated notification service would serve to remind participants via e-mail that their records are due for a refresh, and help to maintain a currency of content imperative to the success of the service. Incorporating a system of rating the quality of records could further incentivise active participation and diligence in maintaining project documentation. Quality assessments would ideally be performed by subject matter experts, but could rely on automated methods based upon record completeness as the next best option if resources for the former are unavailable. And of course, given the GI focus of the service, a web-mapping interface for locating and identifying projects of interest would be a natural extension to the functionality of the original site.

As for the GI Strategy for Scotland as a whole, progress in other areas continues to be made, with ongoing efforts to promote the benefits and coordinated provision of

GI data and services gaining significant traction amongst the wider Scottish geospatial community. And while the Strategy – and the Project Registry – are primarily intended for this Scottish audience, both have and shall continue to be developed with a view to encouraging participation from the wider UK community, and beyond.

References

Annoni, A. and Smits, P., 2003. Main problems in building European environmental spatial data. *International Journal of Remote Sensing*, 24 (20), 3887-3902.

Bédard, Y., Jeansoulin, R. and Moulin, B., 2007. Towards spatial data quality information analysis tools for experts assessing the fitness for use of spatial data. *International Journal of Geographic Information Science*, 21 (3), 261-282.

Bertolotto, M., Carswell, J., McLoughlin, E., O'Sullivan, D. and Wilson, D., 2006. Using sketches and knowledge bases for geo-spatial image retrieval. *Computers, Environment and Urban Systems*, 30 (1), 29-53.

Bilasco, I., Gensel, J., Villanova-Oliver, M. and Martin, H., 2007. Towards Geospatial Queries in a Semantic Digital Library for 3D Data. *Transactions in GIS*, 11 (3), 337-353.

Comber, A.J., Fisher, P.F. and Wadsworth, R.A., 2007. User-focused metadata for spatial data, geographical information and data quality assessments. *In*: M. Wachowicz and L. Bodum, eds. *10th AGILE International Conference on Geographic Information Science*, 8-11th May 2007. Aalborg, Denmark: Aalborg University, p.p. 13.

Crompvoets, J., Bregt, A., Rajabifard, A. and Williamson, I., 2004. Assessing the worldwide developments of national spatial data clearinghouses. *International Journal of Geographical Information Science*, 18 (7), 665-689.

Devillers, R., Bédard, Y. and Jeansoulin, R., 2005. Multidimensional Management of Geospatial Data Quality Information for its Dynamic Use Within GIS. *Photogrammetric Engineering & Remote Sensing*, 71 (2), 205-215.

Devogele, T., Parent, C. and Spaccapietra, S., 1998. On spatial database integration. *International Journal of Geographic Information Science*, 12 (4), 335-352.

Easton, C., 2005. One Scotland–One Geography: A Small Country with Big Ideas. *In*: K. Fullerton ed. *The 11th EC GI & GIS Workshop ESDI: Setting the Framework*, Sardinia, Italy 29 June – 1 July 2005.

GIgateway, 2003. Discovery Metadata Specifications. London, UK: AGI, p.p. 70.

Goodchild, M.F. and Haining, R.P., 2004. GIS and spatial data analysis: Converging perspectives. *Papers in Regional Science*, 83 (1), 363-385.

Harvey, F. and Tulloch, D., 2006. Local-government data sharing: Evaluating the foundations of spatial data infrastructures. *International Journal of Geographic Information Science*, 20 (7), 743-768.

Mathys, T., 2004. The Go-Geo! Portal Metadata Initiatives. *In*: A Lovett, ed. *Proceedings of the Geographical Information Science Research UK Conference (GISRUK) 12th Annual Conference*, 28-30th April 2004. Norwich, UK: University of East Anglia, 148-154.

Movva, S., Ramachandran, R., Li, X., Khaire, S., Keiser, K., Conover, H. and Graves, S., 2005. Syntactic and semantic metadata integration for science data use. *Computers and Geosciences*, 31 (9), 1126-1134.

Rhind, D., 1997. Overview of the National Geospatial Data Framework. *In Proceedings of the Association for Geographic Information – GIS Conference 1997 (AGI - GIS'97)*. London, UK: NGDF, 1-14.

Tait, M., 2005. Implementing geoportals: applications of distributed GIS. *Computers, Environment and Urban Systems,* 29, 33-37.

Automating Geospatial Metadata Generation – An Integrated Data Management and Documentation Approach

Research Paper III

Abstract

Geospatial metadata have long played an important role in the management of geospatial datasets. Often employed by institutions to organise, maintain and document their geographic resources internally, metadata may also provide a vehicle for exposing marketable data assets externally when contributed to on-line geospatial exchange initiatives. In spite of the numerous benefits it affords, obstacles to the production of such geospatial surrogates are numerous. The current work proposes an approach aimed at reducing the effort associated with geospatial metadata generation through the customisation of a proprietary GIS. By coupling data preparation, management and documentation approaches with such a bespoke application, it is intended to mitigate impediments to geospatial metadata generation whilst promoting a system of data administration that safeguards the data it supports. The current prototype, implementing an extended Dublin Core geospatial profile of twenty-three elements, was capable of generating a total of twenty basic metadata entries. While the findings do not suggest a dispensability of human mediation in the authoring process, they do support the view that a dataset's ambient computing infrastructure has the potential to play a significant role in automating the creation of geospatial metadata.

Introduction

Since their appearance in the latter half of the twentieth century, the proliferation of Geographical Information Systems (GIS), their applications and related technologies has continued apace (Goodchild and Haining 2004). With the more current developments in the realm of web-enabled geospatial services, as well as the emergence of popular Geographical Exploration Systems (GES) such as *Google Earth* and Microsoft's *Virtual Earth*, increasing numbers of people continue to be introduced to the possibilities afforded by such technologies. Whether for public, private or academic purposes, demand for geographical information (GI) has in addition increased several-fold in recent times, with those looking to procure data turning to the exploration of existing data pools, commissioning the collection of new data, or resorting to producing their own. Efforts to meet this demand contribute to the "explosion" of data currently available, introducing further problems relating to the management of quite often voluminous data holdings, and how such assets can be successfully exploited[19].

As more data and information are produced, the more vital approaches become for managing and locating such resources (Göbel and Lutze 1998); the role geospatial metadata assumes here has been widely acknowledged[20] (Kim 1999, Tsou, 2002, Limbach *et al.* 2004). Apart from providing a means of documenting a dataset's key statistics such as its quality, appropriateness, currency or area of coverage, metadata

---

[19] Yuan, M., Buttenfield, B., Gahegan, M. N. and Miller, H., 2001. Geospatial Data Mining and Knowledge Discovery. http://ags.ou.edu/~myuan/papers/mining.pdf

[20] Kacmar, C., Jue, D., Stage, D. and Koontz, C., 1995. Automatic Creation and Maintenance of an Organizational Spatial Metadata and Document Digital Library. http://csdl.tamu.edu/DL95/papers/kacmar/kacmar.html.

can supply information on the availability of the data it describes, how it may be accessed and exchanged; it contributes towards data management efforts by helping organise, maintain and locate data resources; when collated into catalogues, metadata collections can be indexed for rapid query, contributed to data clearinghouses or similar data exchange initiatives where they can be used to externally expose marketable data assets; it aids in the coordination of data procurement efforts by raising the awareness of extant datasets, thereby avoiding duplication of effort, redundant storage and obscuring search results.

Further incentives for its use arise when the implications of neglecting metadata entirely are considered. Some claim that the cost of not creating metadata can outweigh that of authoring it, citing concerns associated with employee turnover, data redundancy, conflicts and inappropriate decision-making[21]. Others meanwhile go so far as to argue that data is rendered useless in the absence of any metadata (Qi *et al.* 2004). Despite the arguments, obstacles to the adoption of metadata practices remain. Many view its generation as monotonous and time-consuming, a labour-intensive process that is a major undertaking in itself (Guptill 1999, West-Jr. and Hess 2002), resulting in a pervasive outlook which shuns metadata creation (Mathys 2004). Streamlining conventional authoring processes, and thereby conserving associated resources, would mitigate the barriers to data documentation.

---

[21] Deng, Y., 2002. The Metadata Architecture for Data Management in Web-based Choropleth Maps. http://www.cs.umd.edu/projects/hcil/census/JavaProto/metadata.pdf

The negative perceptions of metadata practices can persist even once they have been adopted, often with harmful consequences for the quality of output. Even where its value is recognised, data documentation commonly takes low priority in relation to other activities, reduced to being seen as “a necessary evil”[22]. And as conventional geospatial dataset documentation remains a largely manual process, it tends not only to be tedious when finally undertaken, but also error prone (Leiden *et al.* 2001). Considering that large volumes of data currently on offer emanate from those not traditionally considered to be geospatial data producers[23], questions arise as to whether the accompanying metadata (when present) consistently reflects that which it purports to document.

The current work proposes an approach aimed at reducing the effort associated with geospatial metadata generation. Further, by combining data preparation, filing and documentation workflows within a combined framework, barriers to the creation of geospatial metadata can potentially be lowered while simultaneously enforcing a system of data organisation designed to safeguard such assets. Regardless of application domain, it is contended that facilitating the accelerated location, retrieval and interpretation of an organisation’s data holdings through the use of metadata can serve to realise the potential of (frequently underexploited) geospatial resources.

[22] Vermeij, B., 2001. Implementing European Metadata Using ArcCatalog - ArcUser Online. http://www.esri.com/news/arcuser/0701/metadata.html.
[23] Schweitzer, P. N., 1998. GIS and Metadata - Putting Metadata in Plain Language. www.geoplace.com.

Related work

Digital library and information science community

Given the rapid and continual growth of accessible digital resources observed since the advent of the World Wide Web, it is unsurprising that efforts to facilitate effective information location, navigation and retrieval through resource documentation have followed. The digital library and internet cataloguing arenas have hosted a number of research initiatives investigating automated metadata generation, motivated by the view that it is "unrealistic to depend on traditional humanly-generated metadata approaches" when considering the volumes of resources involved (Greenberg *et al.* 2006).

Greenberg (2003) elaborates a framework for metadata generation for online content, noting the part standards play in guiding metadata authoring in addition to the roles of human and computing resources. Automated practices therein are categorised into those which employ resource content indexing i.e. are not predicated on the presence of recognised metadata elements, and those employed by commercial search engines, whether using pre-formed metadata or that produced at run-time. Liddy *et al.* (2001) suggest that such automated techniques can produce reasonable results in certain circumstances; Anderson and Pérez-Carballo (2001) maintain that automated methods tend to be more efficient, consistent and inexpensive than human ones. Whatever the proposed method, most agree that automated and manual approaches combined promise the most in producing quality resource documentation (Craven 2001, Greenberg 2004).

The geospatial domain - geospatial data

The very nature of geospatial data dictates a somewhat different approach than those mentioned above. GI data tend to be both highly structured and manifested in a variety of forms, characterised by the presence of some treatment for geometry. Storage techniques vary – even within proprietary systems – from hybrid models that store spatial and attribute information separately across different files to integrated strategies employing relational databases (Batcheller *et al.* 2007). Most geospatial storage formats therefore do not lend themselves to the same probing operations as used with textual resources given their content's relative lack of accessibility.

The lack of sophisticated support for metadata within pioneering geospatial storage strategies meant that any important information not encoded within a dataset needed to be documented elsewhere. Externalising metadata in discrete text files not only bypassed the need for opening often large documents in their host applications when certain dataset attributes were sought; it would also enable the use of existing indexing and cataloguing techniques for data location and management previously mentioned. Authoring tools consisted of common text editors with metadata often recorded in ad hoc or institution-specific conventions with few common guidelines and little provision for interoperability.

Geospatial metadata standards

More recently, geospatial data documentation efforts have been underpinned by the use of standards, viewed as important keys in facilitating metadata exchange, interpretation by individuals and manipulation by machines. Rising from the initial foundations laid by early data standard initiatives (Moellering 1992), geospatial metadata standards, like their mainstream counterparts, aim to define metadata content and structure. Content standards describe a “common set of terminology and definitions for the documentation of digital (geospatial) data” (FGDC 1998), while metadata encoding standards, commonly implemented using XML Document Type Definitions (DTD) and XML Schemas, outline how content is manifested digitally. Used in tandem, these standards in many ways simplify metadata generation, offering a template for content as well as providing guidelines for permitted input.

Considering the detail to which common geospatial metadata conventions are elaborated however, standards can also have a simultaneously detrimental effect, complicating metadata generation and potentially undermining implementation initiatives (Tsou 2002). In the United States, the FGDC’s[24] Content Standard for Digital Geospatial Metadata (CSDGM) of 1998 outlines a standard containing over three hundred data and compound elements (FGDC 1998). The recently ratified ISO 19115:2005 standard for Geographic Information meanwhile details a metadata element set of over four hundred (ISO 2005). Clearly the benefits afforded by full compliance to either standard will be significantly outweighed by the resources necessary to achieve it.

[24] The Federal Geographic Data Committee

Metadata standard profiles have consequently arisen for a variety of application domains. Essentially subsets of a given metadata convention, profiles define a limited set of elements designed for a specialised purpose whilst simultaneously maintaining standard compliance, often simplifying the metadata authoring in the process. Profiles have for instance been developed to enable data location (*discovery metadata*), to help potential users make decisions on a dataset's appropriateness (*exploration metadata*) and to facilitate data utilisation (*exploitation metadata*) (Taylor 2004). Similarly, region-specific profiles abound, such as the Euro-centric ISO profiles overseen by CEN[25] (Longhorn 2005). While evidently useful for reducing the effort of documenting data, profiles on their own make only a minor contribution in improving the issues relating to metadata authoring efficiency.

Geospatial metadata tools

The adoption of standardised approaches to metadata creation, not least the growing popularity of XML, has made the development of generic tools to aid such practices worthwhile. And with the advent of national and international data exchange initiatives[26], increasing focus has been lent towards the development of tools that encourage the production of consistent, conformant metadata.

[25] The European Committee for Standardisation / Comité Européen de Normalisation

[26] Notable examples include the FGDC's National Geospatial Data Clearinghouse in the US and *GIgateway* in the UK

Early stand-alone desktop tools provided the metadata author with an interface for entry, with standards-based output directed to either file or relational database storage schemas (cf. askGIraffe's customised *Microsoft Access* tool (Foy 2001)). Later versions incorporated on-edit or on-export error trapping – using domain lists, DTD or XML schema validation – as well as providing support for metadata parsing, importation and (rudimentary) conversion. Crafted predominantly in the Java or Visual Basic development environments, such tools are characterised by their independence from the proprietary applications commonly used to create and edit the data to be documented. As such, these metadata editors can also function as viewers, permitting the browsing of key (recorded) dataset themes without the need of a GIS suite (West-Jr. and Hess 2002). Examples of such editors include GIS-tec's *Metadata InGeo EntryTool* (Limbach *et al.* 2004) and GIgateway's *MetaGenie Desktop* (Batcheller and Gittings 2006).

Whilst desktop editors may serve to produce metadata for use both within an organisation and beyond, online editors are predominantly used as components of geospatial data clearinghouses and GIS portals as mechanisms for metadata contribution. Used to streamline the submission process, editors such as EDINA's *Go-Geo Metadata Creator* (Mathys 2004) and G-portal's *XML Metadata Resource Editor* (Lim *et al.* 2005) also serve to help reduce metadata redundancy and replication as records are typically edited where they are hosted. In addition, both strategies are often enhanced through the use of context-sensitive help, tool-tips and option lists designed to guide user input and improve metadata quality.

An important trade-off of employing such independent, cross-platform editors is the disconnect of metadata authoring practices from proprietary application workflows. Dataset creation and editing are consequently detached from metadata creation and editing procedures, necessitating diligent update practices involving at minimum two separate applications. Countering this by providing an integrated workflow through which both data and metadata can be maintained will clearly counter this disconnect, thereby minimising the risk of inconsistency.

Geospatial metadata applications

The advantages of leveraging GIS applications to aid data documentation go further than workflow consolidation. Geospatial suites provide largely unhindered access to data stores they support, an important consideration if streamlining metadata authoring through automation is to be achieved. And while more accessible open data formats are not dismissed[27], the majority of data in production environments are held in proprietary data stores – if published market share figures are to be believed[28]. Furthermore, considering inclusion of programming kits within most common GIS applications (in addition to their near-uniform support for disparate data formats) development of bespoke tools is greatly facilitated.

Due to the lack of metadata support amongst the forerunning data stores, early GIS-native tools focussed on metadata extraction or derivation techniques, i.e. where

[27] For example the XML-based Geography Mark-up Language, or GML

[28] Market research firm IDC estimated the market share figure held by open-source desktop software in 2002 at approximately 3% (GIS Monitor, June 12, 2003 - http://www.gismonitor.com/news/newsletter/archive/061203.php)

dataset attributes are mined and transformed for use as metadata items. Approaches commonly involved executing an application script to extract and export dataset information (e.g. projection details and bounding coordinates) to file for subsequent validation. Nevertheless, external text editors were still required for the manual completion of each metadata record produced. A typical example is the FGDCMETA.AML tool written in Arc Macro Language (AML) for ESRI's ArcInfo and designed for use with the FGDC's CSDGM standard.

As the perceived importance of metadata increased, GIS vendors started to introduce enhanced support for metadata both within their software offerings and alongside their data models. Native support for metadata content and schema standards, often manifested as XML, became a standard feature within many software offerings, as did the ability to edit and author metadata in-package. Many dataset properties were now treated as specific metadata items, and as a consequence could now also be harvested directly where before they were derived. In addition, with the near-universal adoption of XML-based technologies and increasing reliance of vendor-specific programming environments on common development platforms such as Sun Microsystem's J2EE and Microsoft's .net, sophisticated turnkey extensions to existing GIS software now became a real possibility.

The present work

The current paper proposes that the efficiency of geospatial metadata generation can be significantly enhanced through proprietary software customisation and considered data preparation. Apart from recognising its position as market leader in the GIS software field (and thereby offering a familiar platform in which to present the current approach) ESRI's ArcGIS was chosen due to its extensive, extensible architecture based on modular programming components (ArcObjects) with which software can be rapidly developed. In addition, its ArcCatalog component "provides a framework for the implementation of a custom metadata environment"[29], and thus presents an existing toolkit with which to build. The development platform employed was Microsoft's .net, chosen to exploit both the platform's support for solution extensibility, but in particular its tight integration with XML (Stephens and Hochgurtel 2002).

A prototype was built in Visual Basic .net and compiled into a dynamic linked library (dll) file that is registered with the ArcGIS application. Although ArcCatalog provides near-uniform access to a range of data storage techniques, a single model was employed to limit the degrees of freedom of the current analysis. The personal geodatabase, an integrated single-user solution based on Microsoft's Access RDBMS technology, was accordingly selected due to the relative ease in which it is configured as well as its positioning between (legacy) hybrid single-user file-based data stores and integrated multi-user database strategies.

The tool is designed to provide an integrated approach to metadata generation, based on a systematic data management structure and facilitating efficient data

[29] Vermeij, B., 2001. Implementing European Metadata Using ArcCatalog - ArcUser Online http://www.esri.com/news/arcuser/0701/metadata.html.

documentation, metadata validation and basic translation. Being native to ArcCatalog it is bound with the dataset initialisation, configuration and management workflow; it may however be readily retooled for use in ArcMap for applications where binding metadata creation with the data editing and analysis workflow is preferred.

On nomination of a dataset within ArcCatalog the tool is initiated via a standard button interface. The user is presented with a metadata editing form which functions as the principal interface of the tool. Pre-formed metadata items held alongside the dataset are instantaneously harvested on form load; elements may either be overwritten manually, completed if absent or selected for revision using the tool's metadata routines. Routines may be run collectively or individually, allowing full user control over the tool's operations. The operations the tool support (illustrated in Figure 1) include:

- Harvesting pre-existing metadata elements generated by ArcCatalog
- Extracting file hierarchy, data and dataset properties and attributes for use as metadata elements
- Harvesting user-prepared metadata templates
- Guiding the visual inspection, modification and completion of metadata records through the structured presentation of record fields on an editing form
- Enabling the importation from and exportation to other standards through the use of a basic metadata crosswalk

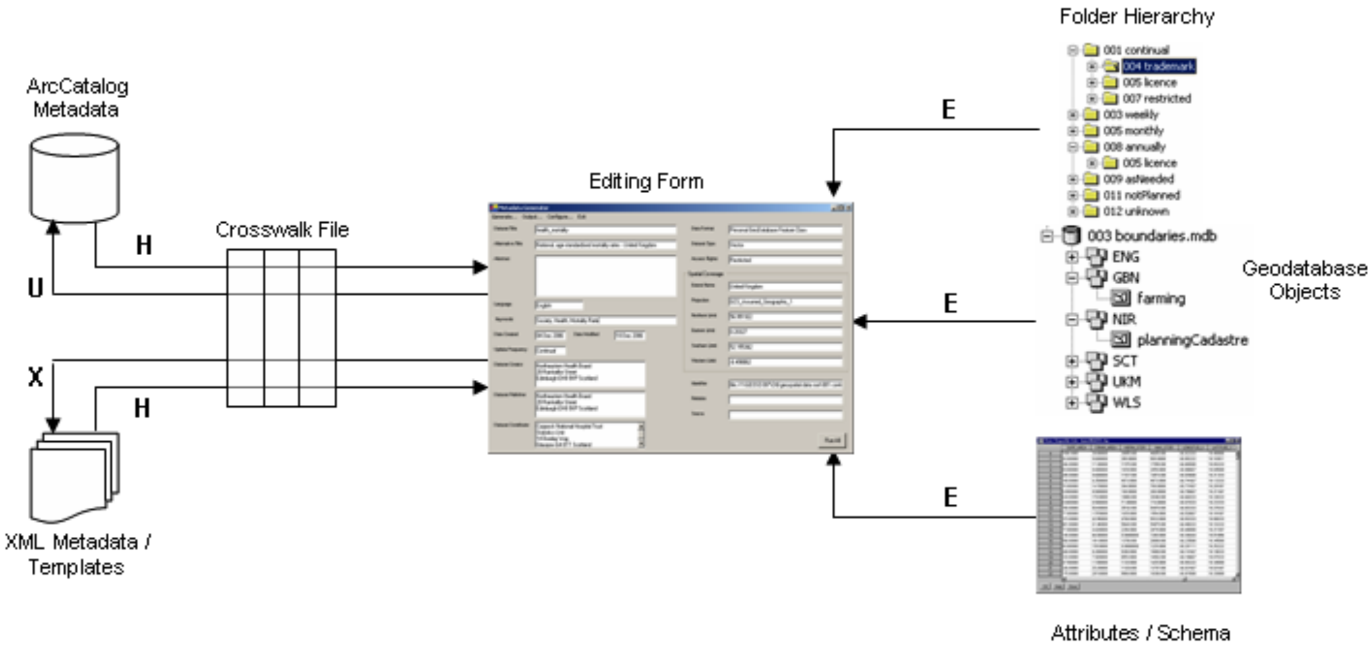


Figure 1. Flow diagram of current metadata prototype, drawing elements from a variety of different sources. H: Harvesting; E: Extraction; X: Export; U: Update.

Metadata standard

A Qualified Dublin Core profile with geospatial refinements was defined to provide a succinct element set with which to test the metadata prototype (Table 1). Offering a widely adopted convention used for depicting any category of resource, Dublin Core was chosen as it affords a sufficiently sparse and manageable solution within the current context. And as a profile of a well defined ISO standard, the set's collection of elements is readily mapped to other more detailed geospatial conventions.

Core Element Name	Element refinement	Description
Title	-	Title
	Alternative	Alternative title
Description	Abstract	A brief narrative summary
Language		Language
Subject	Keywords	Main theme(s)
Date	Created	Date of creation
	Modified	Last date of update
	Period.name	Name of a specific interval. Used here to define frequency of update
Creator		Originating person / organisation
Publisher		Distributing person / organisation
Contributor		Contributing person / organisation
Format		Digital manifestation of resource
Type	Dataset	Nature of content
Rights	Access Rights	Access restrictions
Coverage	Spatial.Box.name	Name of geographic extent of dataset
	Spatial.Box.projection	Spatial reference system of dataset
	Spatial.Box.northlimit	Limits of dataset extent in coordinates
	Spatial.Box.eastlimit	
	Spatial.Box.southlimit	
	Spatial.Box.westlimit	
Identifier		Online linkage to dataset
Relation		A reference to a related resource
Source		A reference to a resource from which the present resource is derived.

Table 1. Qualified Dublin Core element set used in the current work. 15 core elements are qualified by element refinements resulting in a total of 23 fields.

Initial harvesting

Initial harvesting takes advantage of the inherent metadata already collected from registered datasets by ArcCatalog. Stored in ISO-compliant[30] XML alongside the data, prototype routines harvest the required elements contained therein using XPath expressions defined in a custom metadata crosswalk file. Primarily used for cross-mapping conventions as later described, the file details the addresses (in XPath) of elements contained within the dataset's default XML metadata which are retrieved, offering an initial set of fields on which to add. In the current context, eight out of

[30] Specifically, ISO 19115 metadata. Storage in FGDC CSGDSM format is also supported.

the twenty three elements are automatically generated: Title, Language, Date Created, Format, Type, Coverage projection and bounding coordinates as well as Identifier.

Extraction routines

Custom routines are used to extract further information from the dataset, its data content as well as the dataset's location within a refined folder hierarchy. The latter is based on the premise that efficient data management practices employ logically organised data stores. Here, metadata entities are used to organise the very data they describe, providing a nomenclature with which datasets may be labelled, categorised and filed. In the current configuration, personal geodatabases, their contents and the folders in which they reside are tagged according to specific metadata vocabulary terms by which they are unambiguously characterised, facilitating dataset management while contributing to automated metadata record compilation.

The test scenario comprised of a three-level folder hierarchy – a root directory, a primary and a secondary level. Each level denotes a specific metadata element, holding containers labelled using code lists of commonly used ISO standards (Table 2). Personal geodatabases are similarly tagged and stored in a location within the hierarchy which best reflects the attributes of the data within. Personal geodatabase constituents are likewise managed, with appropriate code lists defining how collections of geographic features (feature classes) and their aggregations (feature datasets) are annotated. The test hierarchy is illustrated in Figure 2.

Container	Name	ISO Code List
Primary level	Date Period	19115:MD_MaintenanceFrequencyCode
Secondary level	Access Rights	19115:MD_RestrictionCode
Personal geodatabase	Subject Keyword	19115:MD_TopicCategoryCode
Feature dataset	Coverage Spatial Box Name	3166-2
Feature class	Subject Keyword	19115:MD_TopicCategoryCode

Table 2. Prototype folder hierarchy in which datasets are filed and named. Entire code lists need not be replicated within the hierarchy: containers may be created as required on filing new datasets.

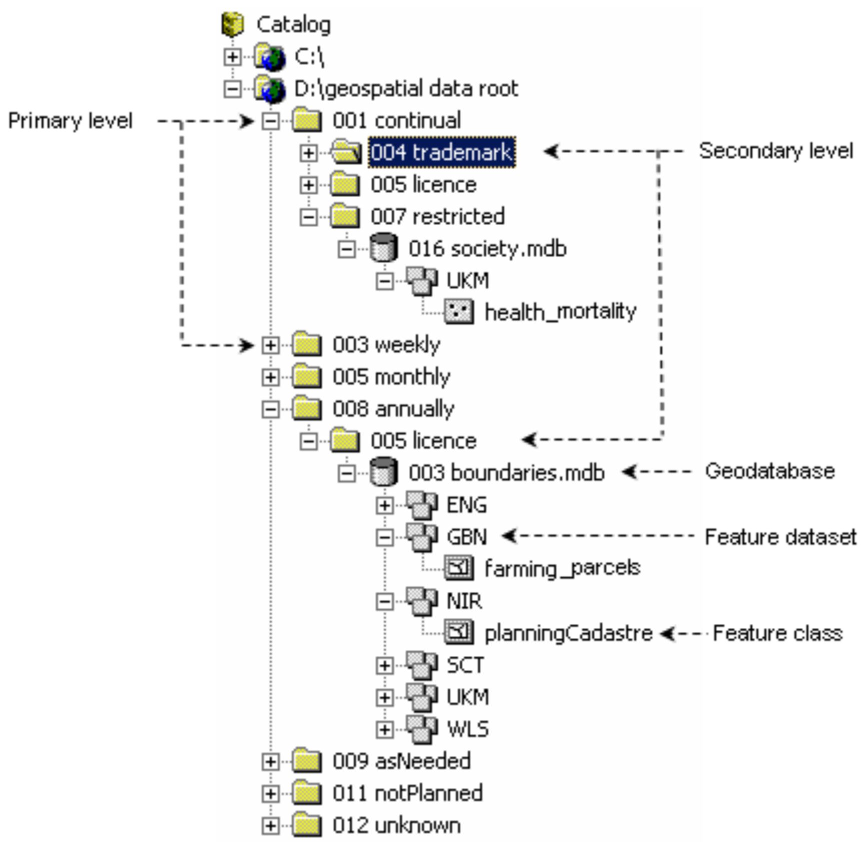


Figure 2. Illustration of prototype data storage hierarchy, yielding five metadata elements. Container tags are assigned on the basis of specific metadata entity code lists detailed in Table 2.

The approach represents a plausible data management protocol which may be readily adapted to different application domains. The entire hierarchy or geodatabase configuration need not be recreated for effective contribution to generating metadata; indeed, a subset of each code list may be preferred, with containers created only when needed to organise an incoming dataset. The contribution of this storage classification strategy is clearly bound to the number of levels in the combined folder and geodatabase hierarchy. Currently, three simple elements (Date Period, Access Rights and Spatial Box Name) and one compound element (depicting two Subject Keywords) are derived.

The method for extracting metadata elements from a dataset meanwhile presupposes that they have been comprehensively compiled. Additional dataset properties, whether supplied by the author or calculated in the process of registering geographic objects, present the opportunity for extending what can be extracted from a data store. Held alongside the data in a similar manner as spatial referencing information such as projection, coordinate system etc, details may be mined using custom code and transformed into usable metadata elements. Currently a single element – Alternative (title) – is created using this extraction method; scope remains to retrieve other elements not currently treated formally as metadata by the program for more detailed standards (including for instance spatial resolution and certain vertical extent attributes).

The routines that extract elements from the data may depend upon a formal attribute schema, or may allow for a relative lack of structure. Indexing frequently occurring textual attributes may for instance serve to extract values that can be adopted as keyword elements; retrieving feature type definitions may nevertheless suffice if a feature catalogue-based schema[31] is adhered to. Date fields may similarly be queried indiscriminately to yield potential maximum and minimum values for a date range of use element, or they may be referenced directly in the event that a predictable data standard is employed. In the present work, a Revision Date element is derived from the contents of a specific field denoting the date of update of each individual feature.

Template harvesting

In a method adopted by many geospatial metadata editors, reusable content may be stored in XML files for harvesting during metadata production. The approach is extended herein through the association of variables managed by the underlying operating system with these pre-prepared templates. Details of a dataset's Creator, Publisher or Contributor for the current Dublin Core profile can for instance be automatically incorporated within each metadata record on the basis of variables such as the current username or the domain of the user's workstation. XML template constituents are again addressed using XPath expressions and retrieved in a similar manner as the method for inherent metadata above.

Metadata editing interface

The tool centres on a form interface through which routines are initiated and metadata items are edited, with the form's fields corresponding to the content of the

[31] ISO 19109:2005 Geographic Information – Rules for Application Schema standard for instance allows for the definition of conceptual data models which define the logical structure of an application's data, commonly instantiated using feature catalogues that define permissible feature types

chosen metadata standard (Figure 3). Inherent metadata elements retrieved through initial harvesting are used to pre-populate the form, such elements being the most up-to-date and which are typically not edited. Extraction and harvesting routines can either be performed during initial pre-population, or manually executed once the form has loaded on-screen. Here, standard elements (for which metadata generating procedures may be applied) are selected in either their entirety or in any specified combination prior to initiation. Metadata items retrieved may be edited and additional ones supplied in an interface complete with similar content guiding mechanisms as those of the stand-alone editors mentioned previously.

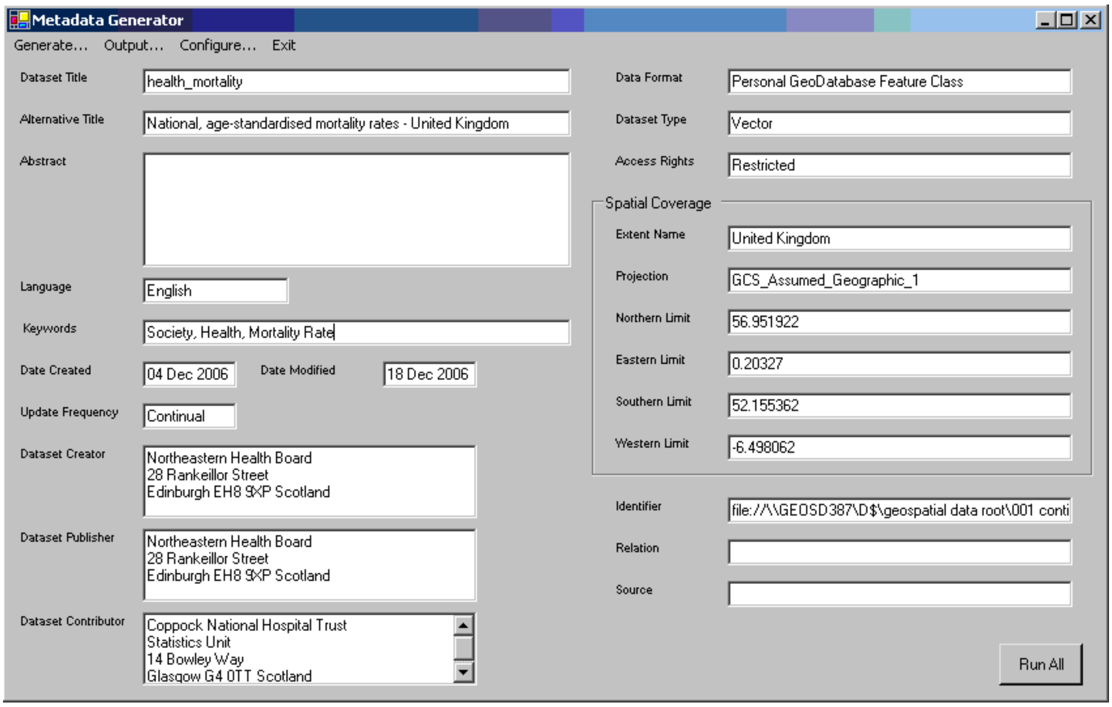


Figure 3. Prototype's central interface. Routines may be run selectively via the Generate / Output menu items or collectively via the Run All button. Routines and elements may also be selected / deselected (via the Configure menu), enabling mediator customisation of the Run All operation set.

Record validation

The method for validating completed metadata records will depend upon the application for which they are intended. Records destined for publication on a geospatial data clearinghouse service may demand strict quality control to ensure that both the XML output is well-formed and the content values are within allowable ranges; the quality demands on metadata produced for internal organisational are frequently less formal and may simply require rudimentary content validation. XML Schema validation in the first instance may be facilitated with the support of the Microsoft XML Core Services (MSXML) that accompany the .net platform. Metadata records produced to meet a specific standard are output to XML where they may be tested for compliance (using methods provided by MSXML) against a corresponding XSD schema registered with the tool. As an alternative to potentially complex schema validation, metadata editor fields may be verified prior to export to file using integrated spell-checking, domain value look-ups and other integrity measures such as verification of mandatory field completion.

Metadata output

One of the key components of the tool is the standard mapping or crosswalk file. Not only used to support the aforementioned metadata harvesting techniques, the file also provides a means of cross-referencing metadata standard elements. Each row in the mapping table denotes a metadata element; columns denote each specified metadata standard. Field values are in the form of XPath references which are used to read from and write to XML metadata. Once the metadata editing form is complete, elements can be written back to the ArcCatalog-native metadata for association with the active dataset or exported to external XML files according to the standard(s) included in the crosswalk file.

Results

The prototype offers a reasonable saving in effort for the metadata producer, albeit measured in the number of elements automatically populated and not actual time savings. Of the total twenty-three metadata standard elements outlined in Table 1 above, twenty were automatically generated using the prototype method. Abstract fields typically demand some degree of intellectual forethought as to their contents; no direct attempt was therefore made to address this element. Other metadata items such as Keywords could nevertheless be used to seed the Abstract entry and guide the manual composition of the said description. Relation and Source elements were similarly omitted as the test data was neither derived nor necessarily related to other existing resources; such elements could however be derived from other feature classes within the same feature dataset (Relation) or from the collection of lineage information if specifically supported (Source).

With regards a dataset's physical location and its impact on element extraction, this will be bound to the value of *n*, or the number of contributing levels in the adopted folder hierarchy. Increasing values of *n* will increase metadata element contribution but potentially complicate dataset classification and storage; the *n* value adopted should offer a compromise between maximising element contribution and the need to retain navigability of the storage hierarchy. A value of two was deemed appropriate for the current investigation – arguments for exceeding this in any future implementation should be carefully considered.

Feature datasets, employed to enable topological operations and impose a further degree of data organisation, must contain feature classes falling within the same geographic region and were therefore considered ideal for binding with elements depicting spatial coverage. While no similar match presented itself for feature classes, the use of the Keyword element was deemed a reasonable fit in the present context. Extending the involvement of hierarchical geodatabase components to metadata generation beyond that made by feature datasets and feature classes was not pursued. Nevertheless, internal aggregations of features supported by the geodatabase's object-relational model could conceivably be leveraged to contribute, and is consequently noted.

Initial harvesting of inherent metadata and the extraction of data properties meanwhile do demand extra effort and diligence when it comes to data preparation, but offer the additional benefit of enhancing data quality. Whilst the quality of inherent metadata by and large depends upon appropriate dataset initialisation (registering the correct projection details for example), data property extraction relies on the completion of dataset variables which may or may not be required within an organisation's application domain. Any decision to cater for such variables will depend on whether it is intended to use metadata to market data resources and whether the adopted metadata standard supports an equivalent element, all the while bearing in mind that an explicit declaration of 'no value' for a metadata element eliminates the uncertainty blank entries present.

Despite the success of querying and indexing techniques employed on unstructured attribute tables, it is suggested that element extraction is better facilitated if such tables conform to a standard data schema whose constituents can be consistently and reliably referenced. Here, extra implementation costs can be partially offset through the incorporation of standard-compliant data dictionaries within surveying equipment, however querying and indexing routines may be preferred in instances where spatial data standards are deemed too unwieldy to implement. A further alternative would be the introduction of a more lightweight feature-level metadata standard on which to base extraction techniques.

Discussion

An approach for enabling the rapid production of geospatial metadata has been proposed, one which demonstrated the potential opportunities geospatial data, their applications and management practices present when undertaking automated metadata generation. While the desire is to automatically populate the maximum number of elements per metadata standard used, the role of human mediators for the purposes of quality evaluation should not be overlooked. As the approach focuses on more than purely metadata record completion, that is, it is also predicated on good data preparation and management practices, there are more potential points of entry for errors, with the subsequent need for extra diligence during data creation.

It could be argued that what is presented here is not so much the automatic generation of metadata but the transfer of effort from metadata production to data management. While this is certainly, but not exclusively, the case, it is proposed as a sound metadata management model as it encourages a well defined data storage scheme and good data preparation; it frees up authoring resources which may now be applied to descriptive metadata and quality control – a conspicuous benefit in cases where data documenters and data authors are distinct; it safeguards metadata quality (contingent on appropriate dataset categorisation and data preparation) as elements are retrieved not entered manually; and presents a significant net saving of time despite the potentially high initial investment. Geospatial metadata has long been advocated to facilitate the management of data collections; the current approach takes this one step further, using metadata standard elements to plan data filing and in the process, contribute to metadata production. And although the test configuration may be potentially perceived as construed, it is argued that data management, by definition, should adhere to a formalised, predictable structure to best facilitate data categorisation and location.

While the present tool demonstrates a functional set of options for the automatic generation of metadata, potential for extension remains. Aside from adapting it to incorporate support for the multi-user geodatabases prevalent in corporate GIS environments, further scope may exists for deriving metadata elements from both the data and the host systems. Geographic extent names can be calculated from the data's geometry by overlaying it with a reference place-name and boundary dataset bundled with the tool. Scripts which track data transformations and changes at the file level may be integrated with existing provision for monitoring data provenance to produce more detailed information on a dataset's pedigree. For organisations participating in data exchange initiatives, the tool can be extended to export an interoperable version of its data along with its metadata. By coupling this output with existing metadata serving software and an open source mapping server, an inexpensive means of visualisation can be supported for organisations wishing to market their data holdings.

Conclusions

It has not been the intention to laud a specific proprietary software offering, or proprietary GIS in general, merely to present what has been possible to achieve extending the basic functionality one offering provides in facilitating the production of quality geospatial metadata. Whether adopted in their entirety or in piecemeal fashion, it is contended that gains are to be had in tackling metadata generation bottlenecks and data management issues with approaches based on those outlined above.

Employing ArcGIS, the coupling of metadata production with dataset workflows was enabled, as was the exploitation of existing programming frameworks for development. While open source GIS offerings have made great strides in recent times – particularly in the realm of data sharing and visualisation across the web – the lack of a mature, widely-adopted production-level desktop GIS with sophisticated metadata support argued against their use. Such a lack may be viewed as curious considering the weight of support behind geospatial data exchange efforts dealing with standardisation, communication protocols and software, particularly considering the pivotal role metadata plays therein.

Considering the attention paid to such data sharing initiatives, one can easily get the impression that the actual generation of quality data surrogates is taken for granted; evidence from the UK's national metadata service GIgateway would suggest that this is in fact one of the major obstacles to geospatial data exchange. Obligating and incentivising the supply of geospatial metadata will only ever work up to point; the key is to encourage a change in mentality towards the process of documenting data. Automating metadata production can help facilitate this, and supports the call to expand the scope of GI exchange efforts to include geospatial metadata generation.

The continuing convergence around ISO standards can contribute here, providing a common way to encode and manipulate metadata, thus increasing the scope for interoperability amongst emerging metadata solutions. The aim should be the development of a generic tool, readily adaptable to any application domain, yet capable of easing the burden of metadata authoring irrespective of data strategy employed. Existing provision for interoperability in proprietary solutions using

OGC-compliant strategies could be extended for this purpose through the support of standard APIs to allow full access to data stores regardless of origin. GML may present an option here; a better approach however would be to access data in its native form without the need for transformation and subsequent potential for data loss.

References

Anderson, J. D. and Pérez-Carballo, J., 2001. "The nature of indexing: how humans and machines analyze messages and texts for retrieval: part I: research, and the nature of human indexing." *Information Processing and Management: an International Journal,* 37 (2), 231-254.

Batcheller, J.K. and Gittings, B.M., 2006. Avenues for developing the UK's National Geospatial Metadata Service. *In*: G. Priestnall and P. Alpin, eds. *Proceedings of the Geographical Information Science Research UK (GISRUK) 14th Annual Conference*, 5 - 7th April 2006. Nottingham, UK: University of Nottingham, 259–262.

Batcheller, J. K., Gittings, B. M. and Dowers, S., 2007. The Performance of Vector Oriented Data Storage Structures in ESRI's ArcGIS. *Transactions in GIS,* 11(1): 47-65.

Craven, T., 2001. DESCRIPTION meta tags in public home and linked pages. *LIBRES: Library and Information Science Research* 11 (2) [online] Available from: http://libres.curtin.edu.au/LIBRE11N2/craven.htm
[Accessed 29 March 2009]

FGDC, 1998. FGDC-STD-001-1998. Content standard for digital geospatial metadata. Reston, VA, USA: Federal Geographic Data Committee, p.p. 90.

Foy, F. B., 2001. *Metadata made easier? Development of improved online tool for "askGIraffe"*. Thesis (MSc). University of Edinburgh.

Göbel, S. and Lutze, K., 1998. Development of meta databases for geospatial data in the WWW. *In 6th International Symposium on Advances in Geographic Information Systems, ACM-GIS '98*, 6-7 November 1998 Washington D.C., USA. New York, USA: ACM, 94-99.

Goodchild, M.F. and Haining, R.P., 2004. GIS and spatial data analysis: Converging perspectives. *Papers in Regional Science*, 83 (1), 363-385.

Greenberg, J., 2003. Metadata Generation: Process, People and Tools. *Bulletin of the American Society for Information Science and Technology,* 29 (2), 16-19.

Greenberg, J., 2004. Metadata Extraction and Harvesting: A Comparrison of Two Automatic Metadata Generation Applications. *Journal of Internet Cataloging,* 6 (4), 59-82.

Greenberg, J., Spurgin, K. and Crystal, A., 2006. Functionalities for automatic metadata generation applications: a survey of metadata experts' opinions. *International Journal of Metadata, Semantics and Ontologies,* 1 (1), 3-20.

Guptill, S. G., 1999. Metadata and data catalogues. *In:* P. Longley, M. F. Goodchild, D. J. Maguire, and D. W. Rhind, eds. *Geographical Information Systems*. Chichester, UK: Wiley, 677-692.

ISO, 2005, BS EN ISO 19115:2005. Geographic Information - Metadata. Failand, Bristol, UK: BSi British Standards.

Kim, T. J., 1999. Metadata for geo-spatial data sharing: A comparative analysis. *The Annals of Regional Science,* 33: 171-181.

Leiden, K., Laughery, K. R., Keller, J., French, J., Warwick, W. and Wood, S. D., 2001. A Review of Human Performance Models for the Prediction of Human Error. Moffett Field, California, USA: National Aeronautics and Space Administration, p.p. 125.

Liddy, E. D., Sutton, S. A., Paik, W., Allen, E., Harwell, S., Monsour, M., Turner, A. and Liddy, J., 2001. Breaking the metadata generation bottleneck: preliminary findings. *In:* E. Fox and C. Borgman eds. *Proceedings of the 1st ACM/IEEE-CS joint conference on Digital libraries*, Roanoke, Virginia. New York, USA: ACM Press, 464.

Lim, E.-P., Liu, Z., Yin, M., Goh, D. H.-L., Theng, Y.-L. and Ng, W. K., 2005. On Organising and Accessing Geospatial and Georeferenced Web Resources using the

G-Portal System. *Information Processing and Management: an International Journal,* 41 (5), 1277-1297.

Limbach, T., Krawczyk, A. and Surowiec, G., 2004. Metadata Lifecycle Management with GIS Context. In *10th EC GI & GIS Workshop, ESDI State of the Art,* 23-25th June 2004 Warsaw, Poland, p.p. 10.

Longhorn, R. A., 2005. Geospatial Standards, Interoperability, Metadata Semantics and Spatial Data Infrastructure. *In Proceedings of the NIEeS Workshop on Activating Metadata* 6-7 July, Fitzwilliam College Cambridge, UK, p.p. 23.

Mathys, T., 2004. The Go-Geo! Portal Metadata Initiatives. *In*: A Lovett, ed. *Proceedings of the Geographical Information Science Research UK Conference (GISRUK) 12th Annual Conference*, 28-30th April 2004. Norwich, UK: University of East Anglia, 148-154.

Moellering, H., 1992. Opportunities for Use of the Spatial Data Transfer Standard at the Sate and Local Levels. *Cartography and Geographic Information Systems,* 19 (5), 332-334.

Qi, L., Lingling, G., Feng, H. and Yong, T., 2004. A Unified Metadata Information Management Framework for Digital City. *In: Proceedings of IEEE's Geoscience and Remote Sensing Symposium*, 20-24 September 2004 Anchorage, Alaska, USA. Washington D.C., USA: IEEE, 4422-4424.

Stephens, R. and Hochgurtel, B., 2002. Visual Basic .NET and XML. New York, USA: John Wiley and Sons Inc.

Taylor, M., 2004. Metadata - Describing geospatial data. *In:* D.D. Nebert, D. D. ed. *Developing Spatial Data Infrastructures: The SDI Cookbook version 2.0*, 24-38. [online]. Available from:
http://www.gsdi.org/gsdicookbookindex.asp [Accessed 29 March 2009]

Tsou, M.-H., 2002. An Operational Metadata Framework for Searching, Indexing, and Retrieving Distributed Geographic Information Services on the Internet. In M. Egenhofer and D. Mark eds. *Geographic Information Science (GIScience 2002), Lecture Notes in Computer Science Vol. 2478*. Berlin, Germany: Springer-Verlag, 313-332.

West-Jr., L.A. and Hess, T.J., 2002. Metadata as a knowledge management tool: supporting intelligent agent and end user access to spatial data. *Decision Support Systems*, 32, 247-264.

A method for automating geospatial dataset metadata

Research Paper IV

Contributors: Bruce M. Gittings and Robert I. Dunfey

Abstract

Metadata have long been recognised as crucial to geospatial asset management and discovery, and yet undertaking their creation remains an unenviable task often to be avoided. This paper proposes a practical approach designed to address such concerns, decomposing various data creation, management, update and documentation process steps that are subsequently leveraged to contribute towards metadata record completion. Using a customised utility embedded within a common GIS application, metadata elements are computationally derived from an imposed feature metadata standard, dataset geometry, an integrated storage protocol and pre-prepared content, and instantiated within a common geospatial discovery convention. Yielding 27 out of a 32 total metadata elements (or 15 out of 17 mandatory elements) the approach demonstrably lessens the burden of metadata authorship. It also encourages improved geospatial asset management whilst outlining core requisites for developing a more open metadata strategy not bound to any particular application domain.

Introduction

Metadata have been key to the development of geospatial data sharing initiatives since their first emergence. From early mention of 'data registers' of spatial assets in the Chorley Report (Department of the Environment, 1987), through pioneering initiatives such as the Federal Geographic Data Committee's (FGDC) National Geospatial Data Clearinghouse in the US and the National Geospatial Data Framework (NGDF) in the UK, to more recent Internet services such as the *Geospatial One-Stop*, *GIgateway* and the latter's academic counterpart *Go-Geo!*, what has become known as geospatial resource metadata has consistently assumed a central pillar around which efforts have evolved (Nanson *et al.* 1995, Davey and Murray 1996, Göbel and Lutze 1998, Guptill 1999, Tsou 2002, Westbrooks 2004). Predicated on the basis that textual metadata records are more widely accessible to location and retrieval techniques than the geospatial holdings they describe, such services aim to facilitate and promote data exchange by registering such data surrogates within searchable catalogues or clearinghouses. Users poll catalogue contents based on dataset properties depicted in the metadata (e.g. spatial coverage, theme, keyword); result sets inform decisions as to whether datasets should be pursued, and where they may be ultimately located.

While it could be argued that the core function of metadata in driving data sharing efforts has remained largely unchanged from pioneering implementations to date, encoding strategies and the mechanisms through which metadata records communicate dataset availability have seen significant progression. Fuelled principally by a desire for interoperability, domain specific metadata strategies have made way for broader standards at regional and national levels, culminating in the recent convergence around International Organization for Standardization (ISO) conventions (Boxall 2003, Nogueras-Iso *et al.* 2004, Xie and Shibasaki 2007). Similarly, computational frameworks used to enable geospatial resource discovery continue to witness a shift away from applications developed primarily for other disciplines, such as the library community's Z39.50 protocol (Hill *et al.* 1999, Tulloch and Robinson 2000, Tsou 2002) to more recent strategies designed specifically for the geospatial domain, in particular those based upon Open Geospatial Consortium (OGC) specifications. Sophisticated proprietary offerings in addition to those produced by the open source community have arisen, providing geo-centric solutions with the potential to empower geospatial data providers wishing to publicise their holdings while also facilitating end-user discovery (Beaujardière *et al.* 2000, Crompvoets *et al.* 2004, Maguire and Longley 2005, Nogueras-Iso *et al.* 2005, Lutz and Klien 2006).

Further a-field, technologies with the potential for enhancing geospatial data exchange continue to surface. Improved networking infrastructures, coupled with the heightened availability and plummeting costs of high-speed broadband and wireless connectivity have in turn served to streamline the access and transfer of the high data volumes often characteristic of geographic datasets (Buttenfield 2002, Turner 2006, Yang *et al.* 2007). The appearance of Geography Mark-up Language (GML) has helped alleviate many of the concerns relating to data compatibility and interoperability, providing an open dialect for data transfer not bound to specific software offerings (Boucelma and Colonna 2004, Agarwal 2005, Peng 2005). Meanwhile, maturing open web portrayal techniques such as those propounded by the OGC offer scope for integration with current discovery services, allowing spatial data to be previewed prior to committing resources for procurement (Nebert 2004, Dunfey *et al.* 2006). Similarly, emerging geoprocessing services such as those aligned with the OGC's Web Processing Service specification offer the potential for remote processing prior to data retrieval, maximising transfer efficiencies whilst minimising end-user software and hardware demands (Aloisio *et al.* 1999, Fonseca and Egenhofer 1999, Kiehle 2006, Healey and Delve 2007).

And yet, considering the role that geospatial metadata retains in facilitating the discovery and exploitation of the very data resources on which many of the aforementioned services depend, the lack of practical efforts aimed at providing easier methods for its generation is telling. While the OGC rightly focuses on issues of abstract interoperability and interface specifications (OGC 2003), actual implementations of metadata-based specifications are invariably predicated on the assumption that metadata instances will be forthcoming (e.g. United Nations Food and Agriculture Organisation's GeoNetwork). For initiatives that explicitly address metadata generation using browser and desktop-based metadata editors (e.g. GIgateway, Go-Geo!), assistance rarely exceeds guiding manual metadata completion and consequently does little to surmount perceptions of a task as being tedious and of low priority (Higgins *et al.* 2003, Balsoy *et al.* 2005). Some proprietary packages (e.g. ESRI's ArcGIS, Cadcorp's SIS) do incorporate metadata management applications capable of generating rudimentary element sets for given geospatial resources, although these invariably demand significant human input to meet compliance with even the most concise of standards.

It is this need for human input that contributes towards a 'metadata bottleneck' (Greenberg *et al.* 2005) that hinders documentation, whether for resource discovery, management or exploitation. Attempts to address this impedance have to date largely fallen within two camps. Those of the first strive to motivate and mobilise metadata authors (education drives and workshop events) (Guptill 1999, Tulloch and Robinson 2000, Mathys 2004); those of the second assume mobilisation, and focus on steering metadata completion (metadata profiles, guidelines and editors). While the role both strategies play in encouraging data documenting practices can be significant, their impact in diminishing the burden of metadata authorship remains open to question.

This paper proposes a method to mitigate this burden through the automation of metadata authoring process steps. Whether from the perspective of cataloguing internal digital holdings, exposing marketable datasets online via metadata services or fulfilling legal requirements by contributing to national spatial data infrastructures (NSDI), the implications of metadata automation are several. By providing computational support to metadata authors, costs of generation can be lessened whilst rendering the prospect of creation less daunting for those yet to undertake it. Such support also reduces the opportunity for human error during metadata record completion; it enables more records to be produced with equivalent effort; and it facilitates easier update of existing metadata following changes to their underlying data. Further, authoring resources can as a result be released for application elsewhere, whether for more intellectually challenging documentation tasks (e.g. quality control, descriptive metadata) or otherwise. It is thus seen to play an important role in efforts to diminish the stigma associated with metadata and its creation, a critical consideration if documentation practices are to be more widely promoted.

Study approach

One method for automating metadata production is grounded in the premise that a dataset, its contents and ambient computing environment can be leveraged for their contribution. In Batcheller (2008), the approach presented focussed around a proprietary GIS and standard-based data management protocol. Metadata elements were computed from a dataset's position within a folder hierarchy, the dataset construct itself as well as from sources pre-compiled both manually and by the host GIS. Metadata were also extracted from dataset data, but here the approach was limited in two notable aspects – attribute data did not adhere to a formal schema and consequently did not reflect conditions typical of production environments; and no effort was made to mine the geometric component distinctive of geospatial datasets.

In the current paper we present an extension to the original approach, one that expands the contribution of both spatial and aspatial data in generating dataset metadata automatically. We propose that the exertions of manual data documentation can be minimised by assuming a holistic approach to distinct processes such as data modelling, creation, storage and management; the founding work is briefly outlined in accordance with this view. We go on to elaborate an extensible, interoperable schema for structuring and documenting attribute data at the sub-dataset level and detail its novel use in generating higher-order metadata. We introduce the concept of the boundary reference layer, illustrating how it computes metadata items from dataset geometry beyond the conventional provision of bounding coordinates and how the process may mitigate element-specific update latency. Further, we describe the use of published guidelines to support completion of descriptive metadata, the computational creation of which is frequently overlooked due to its perception as an exclusively manual endeavour. Finally, we describe a means for incorporating geospatial asset visualisation for the purpose of decision support.

Software environment

With the assumption that geospatial resource documentation commences in close proximity to the datasets being depicted, the proposed utility was developed for ESRI's ArcCatalog, the data and metadata management component of its ArcGIS 9.1 suite. Incorporating an authoring tool within the same application used for dataset registration and maintenance carries with it a number of advantages: it minimises the data-metadata disconnect at source, alleviating latency concerns; editing and update practices for both data and metadata are consolidated within a single package, enabling workflow integration; and it negates the need for external editors where authoring effort may be duplicated and which presents further opportunity for introducing errors.

The application logic that drives metadata generation falls into one of two categories, as described by Greenberg (2004). Routines that *harvest* gather items already held as metadata within the given domain, yet often dispersed throughout it; those that *extract* must transform data read from resources within the domain into metadata-ready form. Metadata elements are consequently computed from three principal sources – the dataset requiring documentation, its immediate computing environment and pre-prepared content unlikely to vary between datasets of the same origin.

Geospatial metadata standard

The UK GEMINI (GEo-spatial Metadata INteroperability Initiative, a.k.a. GEMINI) convention was selected to instantiate the metadata output of the prototype. As the prevailing discovery standard in the UK it offers a widely recognised, succinct format with a proven pedigree in geospatial production environments, one in which the performance of the present work can be demonstrated. Derived in accordance with ISO 19106 Geographic Information: Profiles from both ISO 19115 Geographic Information: Metadata and the UK eGovernment Metadata Standard (eGMS), GEMINI outlines thirty-two elements (Table 1) that are readily mapped to other ISO-based formats (AGI 2004).

Element	Description
Title	Dataset name
Alternative title	*Alternative name*
Dataset language	Language used
Abstract	Narrative summary describing the dataset
Topic category	Main themes of dataset (high-level categories)
Subject	Topic of the dataset content (low-level categories)
Date	Data capture period
Dataset reference date	Date of dataset publication
Originator	*Originating person or organisation*
Lineage	*Dataset pedigree*
West bounding coordinate	Western limit of dataset
East bounding coordinate	Eastern limit of dataset
North bounding coordinate	Northern limit of dataset
South bounding coordinate	Southern limit of dataset
Extent	Geographic identifier of dataset
Vertical extent information	*Vertical domain of dataset*
Spatial reference system	Name of spatial reference system
Spatial resolution	*Capture precision of data*
Spatial representation type	*Method of spatial representation*
Presentation type	*Method of data manifestation*
Data format	Digital format of data
Supply media	*Method of data supply*
Distributor	Distributing organisation
Frequency of update	Prescribed frequency of data update
Access constraint	*Rights of data access*
Use constraints	*Rights of data use*
Additional information source	*Source of further details about dataset*
Online resource	*Online sources of dataset*
Browse graphic	*Illustrative sample of dataset*
Date of update of metadata	Last date of metadata update
Metadata standard name	*Name of metadata standard and profile used*
Metadata standard version	*Metadata version used*

Table 1. The UK GEMINI 32 metadata element set - optional elements in italics.

Founding work

Preliminary generation – dataset initialisation

Metadata generation commences upon registration of new data within the ArcCatalog application. Properties set by the user during dataset initialisation (e.g. *Spatial reference system*, *Title*, *Bounding coordinates*) as well as those taken from the software's default configuration (e.g. *Dataset language*) are collected by the package and stored alongside the dataset as an XML-based metadata record that is available for subsequent editing, or collection as in the current scenario. Other existent dataset properties not specifically treated as items of metadata by ArcCatalog (e.g. *Alternative title*, *Vertical extent information*) are not accessible in this way and must therefore be targeted by extraction routines run against the dataset.

Geospatial data management protocol

Participating datasets are organised within a multi-tiered folder hierarchy, the members of which are tagged on the basis of the data to be held therein; tag nomenclature is based upon domain instances of entities as defined by the GEMINI standard. Here a two-tiered hierarchy is used to store personal geodatabases, the defining entities of which were chosen for their presumed suitability for categorising and filing datasets (Table 2). Data are further classified within each geodatabase using the ArcGIS feature dataset and feature class constructs, the latter representing the geographic layer to be documented. What results is a virtual, predictable path to the data that not only enables dataset retrieval but also serves to contribute a subset of dataset properties for documentation purposes (Figure 1). Consequent population routines manipulate this path and assign its components to the appropriate metadata fields. The entities used and their domains are detailed in Table 2; a prototype hierarchy is provided in Figure 1.

Data container	UK GEMINI Entity	UK GEMINI Domain
Primary tier	Frequency of update	MD_MaintenanceFrequencyCode
Secondary tier	Access constraints	MD_RestrictionCode
Personal geodatabase	Use constraints	MD_RestrictionCode
Feature dataset	Topic category	MD_TopicCategory
Feature class	Title	Free text

Table 2. Data container definitions based upon those UK GEMINI entities deemed appropriate for dataset categorisation. Actual container instances are labelled using the respective UK GEMINI domain values.

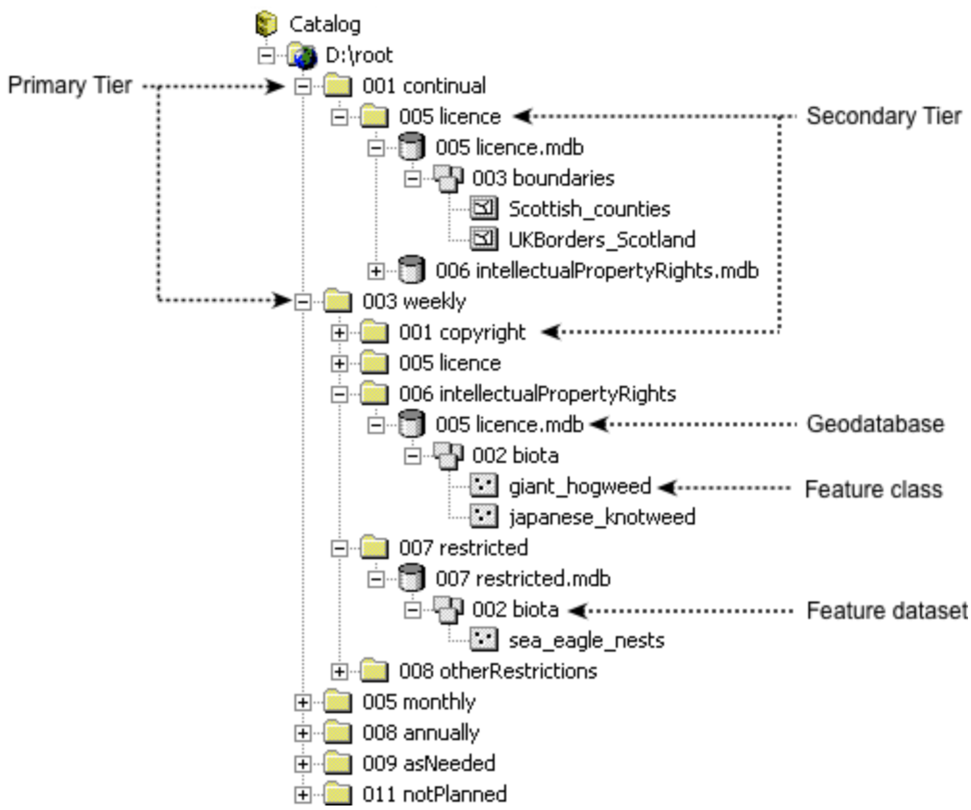


Figure 1. A prototype data container hierarchy, yielding five metadata elements (adapted from Batcheller, 2008). Containers are instantiated on arrival of new datasets; an entire UK GEMINI-based hierarchy need not be constructed.

User-defined content

Geospatial metadata records arising from a single source invariably contain information that either remains constant or changes infrequently across instances, such as the contact details of the data producer. For elements that differ according to user or location (such as *Distributor* details where multiple distribution sites exist), information stored in content templates are loaded on the basis of local operating environment variables such as username or domain. For organisations participating in geospatial data sharing initiatives, default domain values for GEMINI elements such as *Presentation type* and *Supply media* may be set to *mapDigital* and *onLine* respectively – entries that can be hard-coded into any automated approach but which can be overwritten if needed. *Metadata standard name* and *Metadata standard version* elements are similarly initialised, while event related entries such as *Dataset reference date* and *Date of update of metadata* can be automatically time-stamped.

Expansion of forerunning work

Dataset derived metadata – feature metadata

Attribute data have the potential for considerable contribution towards automated documentation given their role in explicating the discrete features that constitute a dataset; tapping this potential is nevertheless complicated by the variety of ways in which attributes may be ordered and encoded. One option is to index commonly occurring values from those semi-structured attribute fields not conforming to specific data standards; as the data are encoded in an inconsistent fashion however, any extraction routine run will necessitate human mediation of each candidate metadata value returned. Spatial data standards impose a predictable structure on attribute schema and can thus address this need for constant review, but their diversity complicates selection even prior to contemplating interoperability-related concerns (Kokla and Kavouras 2001) and the bloat their verbosity can add to data preparation and upkeep.

A more concise alternative for enforcing predictable attribute structure is proposed, one that allows for the association of a metadata schema with the features of each dataset – and in turn, address concerns articulated by Hunter (2001) and Devillers *et al.* (2002) relating to the depiction of geometric primitives. Accordingly, the approach allows for the consistent referencing of member fields whilst providing a mechanism for tracking a dataset's atomic features. Distinct from conventional theme-driven standards, the proposed feature metadata approach was designed to be:

- domain-independent, compatible with existing standard-based techniques and yet easily extensible to permit flexible adoption
- associated specifically with the geometric component of a dataset, thereby disengaging feature metadata from potentially inter-changeable theme-based data and enabling its persistence
- straightforward to apply to existing data models, with potential for automated completion at the point of survey of new geographic features

- capable of contributing towards automating metadata creation at the dataset level without complicating data creation and maintenance

While a review of existing (published) feature metadata strategies did present a number of pre-existing alternatives (most notably the State of Maine's GIS Feature Metadata Recommendation 2000[32]), none met all of the above conditions. As a consequence the current paper elaborates a course more closely aligned with ISO-based practices, heeding the stated developments in the standards community.

ISO 19115:2005 Geographic Information – Metadata (ISO 2005a) for instance details a content schema for the documentation of geospatial data, but its focus lies with the depiction of geospatial resources at the dataset, and to a lesser degree, dataset series level. Metadata at the sub-dataset level are presented within a metadata hierarchy, but suggested implementations only include definitions at these levels when exceptions occur. Further, as metadata conforming to ISO 19115 are held discrete from the resources they describe by convention – regardless of its granularity – this treatment for metadata is on its own insufficient when reviewed for the current purpose.

The ISO 19109:2005 Geographic Information – Rules for Application Schema standard meanwhile allows for the definition of conceptual data models that define the logical structure of an application's data (ISO 2005b). Geographic feature types

[32] http://apollo.ogis.state.me.us/standards/flmeta/fmrecommend.htm

are classified based on a structure defined by the General Feature Model (GFM); feature type definitions (detailing feature attributes, operations and association roles) may be elaborated in feature catalogues. Of particular interest is its specific treatment for feature attributes as well as the general ability to integrate any ISO 19109 application schema with other ISO standard schemas. Here, any feature attribute (GF_AttributeType) can have atomic metadata items associated with it by sub-classing entities beneath the GF_QualityAttributeType specialisation of the GF_MetadataAttributeType entity (Figure 2). Attribute types accordingly defined (specifically, to carry feature metadata information such as quality) obtain their value type definitions and value domains from the ISO 19115 MD_Metadata entity.

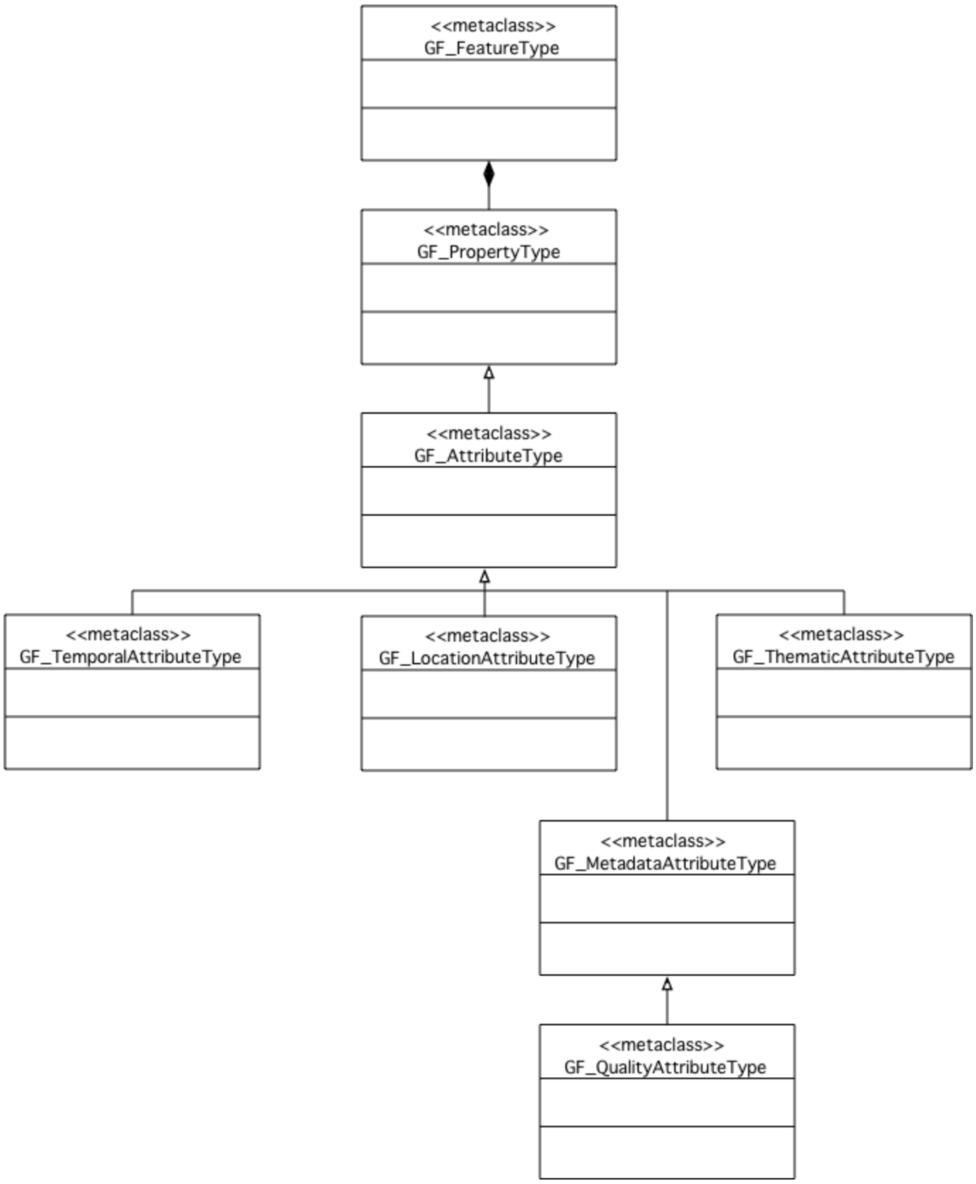


Figure 2. Attributes of ISO 19109 feature types, with an emphasis on the route to metadata-relevant subclasses. Feature metadata instances are catered for by GF_QualityAttributeType's dependency on the ISO 19115 entity MD_Metadata. (Subset taken from ISO 19109:2005).

A six-field element set similar to what can be expected within formal production environments for the purposes of feature-level data tracking was consequently elaborated and mapped using the ISO framework (Table 3). Aliases serve as shorthand field labels, while full element definitions are included within dataset metadata instances by default to permit interpretation. Field domains may be enforced in applications where stringent compliance is required, however these were not currently applied. For new data holdings, the schema may be incorporated at the point of survey through the inclusion of a data dictionary within the surveying equipment, thereby facilitating pre-population prior to registering the dataset with a GIS. Existing datasets may have equivalent entries mapped to the schema if such entries exist; alternatively the schema may be appended to an incumbent attribute table for subsequent completion. Once implemented, fields may be analysed and extracted in accordance to the dataset metadata element for which they may be applied.

Field	Definition	Alias
Originating organisation	DQ_DataQuality.lineage> LI_Lineage.source> LI_Source.sourceCitation> CI_Citation.CI_ResponsibleParty.organisationName	origin
Originating Capture Process	DQ_DataQuality.lineage> LI_Lineage.sourceStep> LI_ProcessStep.description	capture_process
Originating Capture Date	DQ_DataQuality.lineage> LI_Lineage.sourceStep> LI_ProcessStep.dateTime	capture_date
Precision	DQ_DataQuality.report> DQ_Element.DQ_PositionalAccuracy.result	precision
Editing organisation	DQ_DataQuality.lineage> LI_Lineage.processStep> LI_ProcessStep.processor> CI_ResponsibleParty.organisationName	editor
Edit Date	DQ_DataQuality.lineage> LI_Lineage.processStep> LI_ProcessStep.dateTime	edit_date

Table 3. ISO-based feature metadata schema, defined using ISO 19115 entities according to ISO 19109 guidelines (all being subclasses of MD_Metadata.dataQualityInfo). Both organisationName entries (origin, editor) may be replaced with (or supplemented by) individualName if required by the domain of application.

The *origin* field is used to populate the GEMINI element *Originator* in the current prototype. Multiple entries are generalised to indicate the single dominant originator reflecting instances where only one entry is allowed; multiple summary values may also be used where permitted. *Spatial resolution* is calculated from *precision* using the lowest common denominating value of feature precision, although this may also be computed on the basis of average value if preferred. The date range *Date*, used to indicate the data capture period, is calculated with functions identifying the minimum recorded *capture_date* and maximum *edit_date* of features within the dataset. Dataset provenance, as treated by the *Lineage* element to store 'information about the events or source data used in the construction of the dataset' (AGI 2004) is similarly populated. Metadata fields used to track feature history (i.e. *capture_process*, *capture_date*, *editor*, *edit_date*) contribute here, and may be used to append details of each process step to the *Lineage* element.

Dataset derived metadata – geometry

Deriving metadata from data geometry is conventionally limited to the calculation and update of projection-dependent bounding coordinates of the entire dataset. Typically employed to provide a rudimentary spatial component for searching indexed metadata catalogues, such coordinates do not however convey extent information that is easily interpreted by end-users. Boundary names (e.g. political, administrative regions) provide more user-friendly extent attribution but as these are not directly coupled with the spatial component of the dataset, any change to the data's geographic bounds will not be reflected in such attribution unless specifically detected and manually addressed.

Such issues are circumvented through the association with the data a layer depicting application-relevant boundaries, enabling the automated identification of a dataset's potentially evolving extent. Feature sets are programmatically overlain with such boundary reference layers, returning the name of the enveloping region. Extent identifiers within the metadata are accordingly coupled with underlying spatial data, removing the need for manual nomination and update that can be subjective and open to error.

A related issue pertains to spatial resolution. The base GEMINI specification for instance outlines an *Extent* field domain of high level areas such as England, Wales, Great Britain, British Isles, but sole inclusion of a reference boundary layer with these regions may be insufficient for applications with more fine-grained requirements, as with those of local government. Incorporating subsequent reference layers of increasing scale can facilitate improved extent identification and help relay more precise information within the resulting metadata. To illustrate, a test dataset containing the geographic distribution of the invasive Giant Hogweed species (*Heracleum mantegazzianum*) in the south of Scotland is used (Figures 3, a-c). Figure 3a depicts the problem of registering an extent of Scotland for a rather localised phenomenon found in the south east of the country. Introducing an administration boundary layer provides more detail (Figure 3b), showing that the recorded extent is in fact confined to the Scottish Borders. Figure 3c illustrates a hypothetical spread of the weed outwith its original confines and into South Lanarkshire and Dumfries and Galloway; employing such a reference boundary layer not only enables the automatic detection of cross-border spread at run-time, it produces richer metadata on which more informed decisions may potentially be based.

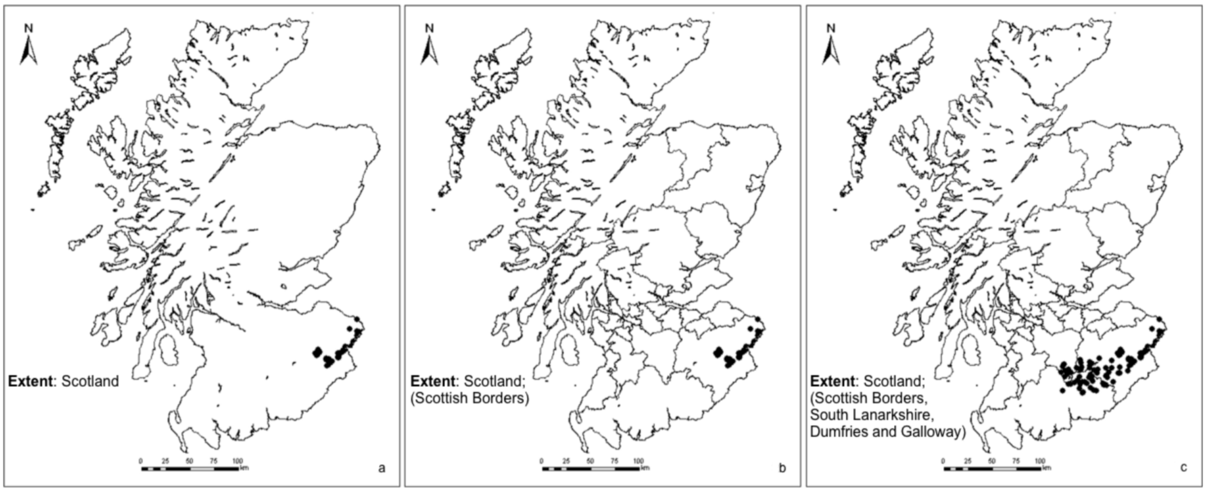


Figure 3(a-b): Geographic extent estimation of *Heracleum mantegazzianum* in the south-east of Scotland, 2003-05, and the effects of boundary reference layer granularity (data courtesy of the Tweed Foundation). **(c):** A hypothetical spread of the species and the districts detected using the boundary reference layer approach.

Abstract seeding

Metadata conventions typically permit the inclusion of free-text resource descriptions via an *Abstract* field that stands in contrast to other domain-controlled elements. While it is perhaps unrealistic to expect metadata automation efforts to adequately complete such narrative entries, the process may nevertheless be assisted through the seeding of abstract fields with the information items from which they are comprised. Guidelines for completing UK GEMINI metadata for instance outline a checklist for abstract completion that demands 13 distinct components, many of which may be extrapolated or approximated from existing (populated) sources (Table 4). Contribution towards abstract creation is thus performed by seed routines that return abstract items into the field for subsequent elaboration manually. Serving not only to mitigate the effort required to complete the element, the approach ensures consistent inclusion of abstract-relevant items that hold the potential of being overlooked where authoring is left unassisted.

Guideline	Seed source
What the dataset depicts	Alternative Title
Area of coverage	Extent
Period of coverage and frequency of update	Date, Frequency of update
Data capture scale / resolution	Spatial Resolution
Data capture method	capture_process
Suggested uses for data	Topic
Category of features depicted	Subject
Details of limitations in data	-
Data linkages	Feature ID field
Data originator(s) / editor(s)	Originator, origin, editor
Data model	Spatial representation type
Data format	Data format
Data series	Sibling feature classes

Table 4. Guidelines for completing abstract content (adapted from recommendations published by GIgateway at http://www.gigateway.org.uk/metadata/standards.html)

Asset Visualisation

While GEMINI was adopted to illustrate the opportunities for automating metadata regardless of its ultimate use, it is perhaps worth considering the specific case of enabling resource discovery. GEMINI elements *Online resource* and *Browse graphic* are of specific interest here; the former enables pinpointing of the resource described, the latter is designed to support decisions based on whether the resource should indeed be pursued. A common approach to address these entries has been to provide a URL for the distributor's website while including a manually generated image snapshot of a subset of the data. The alternative proposed in the current work aims not only to facilitate improved metadata completion but also to provide for more effective visualisation with the potential for fast-tacking access to the data in question.

Using OGC-compliant web portrayal services, datasets can be ‘broadcast’ for immediate use when unlicensed, or identified for subsequent purchase. Layers free of licensing restrictions may be served in accessible Web Feature Service (WFS) format whereas proprietary holdings may be visualised in Web Mapping Service (WMS) format prior to procurement. Service-specific Uniform Resource Locators (URLs) are subsequently written to their associated metadata record, coupling it with the portrayal service.

Execution

The utility was coded using Microsoft’s VB.NET and ESRI’s ArcObjects framework, embedded within ArcCatalog as a dynamically linked library and exposed as a custom button within the application toolbar. Data instantiated as feature classes within the folder-geodatabase hierarchy are selected individually within the application’s file browser and the tool is engaged. Harvesting and extraction routines, initiated from a form interface, together compute metadata elements for display within the form where they are made available for quality control and update prior to final output (Figure 4). Pre-formed metadata items are harvested from ArcCatalog’s XML store using XPath expressions that allow referencing of specific elements. These expressions are read from a rudimentary crosswalk file that not only details the individual element sources within the application's XML metadata repository but also describe (in XPath) where form values will be written to within the XML-formatted GEMINI output document. Initial population routines return all elements in the event of prior completion; further routines may be executed selectively where elements are incomplete or require revision.

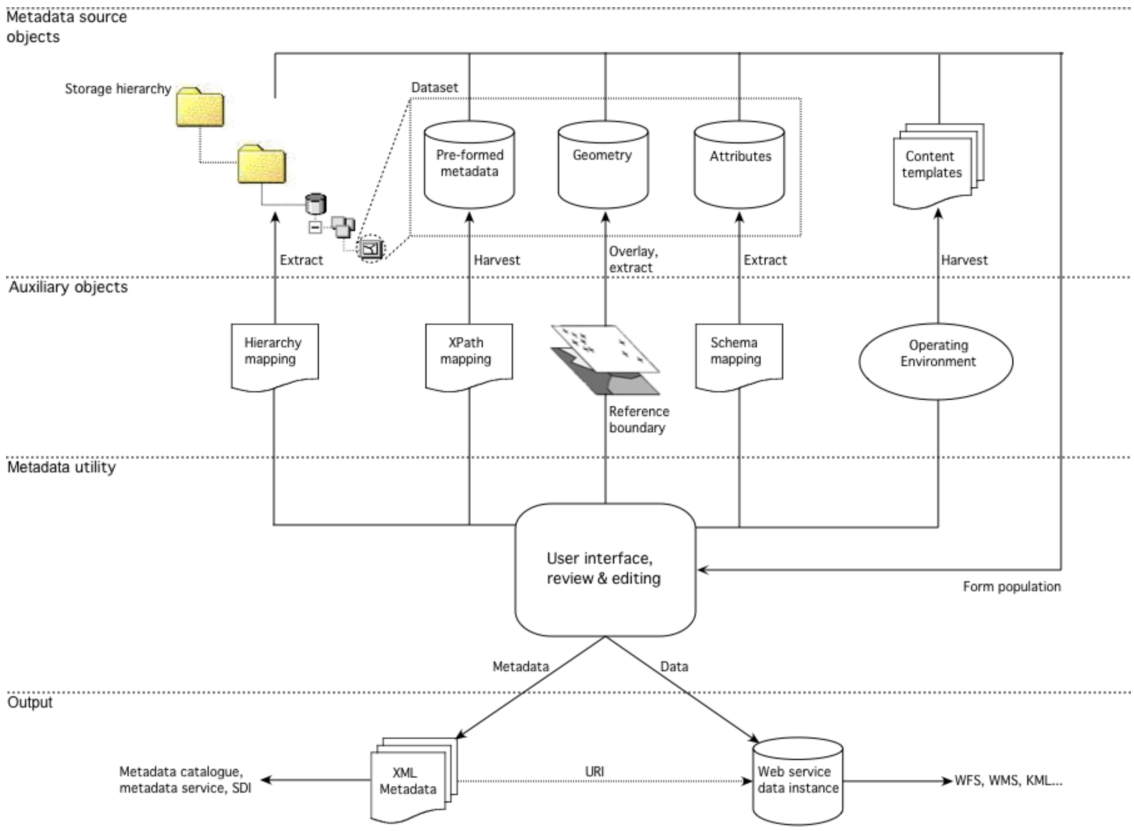


Figure 4. An overview of approach's routines, the metadata sources on which they operated (assisted by its associated auxiliary object). Computed values are displayed on a form interface where they may be modified by hand prior to validation and eventual output.

Elements are computed from feature metadata content at run-time, referenced via their corresponding field names; routines index, summarise and identify maxima and minima according to the GEMINI element they contribute to. Dataset geometry is overlain with the incorporated reference layers to generate boundary names that are then added to the *Extent* field and later used to seed the *Abstract* entry. The application extracts elements from the storage hierarchy by parsing the dataset path and assigning its constituent tokens to form fields according to a path – element lookup table. Pre-authored content templates held within the file system as XML are harvested on the basis of active username, the contents of which are again addressed using XPath. Seeding of abstract content is performed as a penultimate event once other fields have been populated; contributing elements are harvested directly from the form and passed to the abstract field where they are elaborated upon manually.

Metadata records are output as XML files once they have been evaluated and timestamps to actualise the *Date of Metadata update* element have been applied. Metadata passing validation may now be exploited as surrogates for the data they depict, whether locally or following contribution to an off-site repository.

Providing support for visualisation is implemented by combining the metadata output operation with the conversion of the active dataset to an intermediary web portrayal format. Datasets are exported to an Apache Tomcat web container from where they are served as WFS layers by the open source Geoserver application; WMS and Keyhole Mark-up Language (KML) formats may also be configured for visualisation according to the terms of access for the associated data. Following initial service composition, asset-specific URLs are written to their corresponding metadata records; subsequent data updates overwrite existing data serving instances, forgoing the need for further link revision.

Metadata output

Of the seventeen mandatory elements detailed by the UK GEMINI convention, fifteen are automatically populated, the exceptions being the *Abstract* (partial) and *Dataset reference date*, the latter indicating a notional publication date. Twelve further optional elements are generated (two arising from default values hard-coded into the utility) leaving three entries requiring manual completion – these being the elements associated with supplemental dataset description and portrayal: *Online resource*, *Browse graphic* and *Additional information source*. While a total contribution of twenty-seven entries will undoubtedly mitigate the burden of authorship, it should be noted that there is little treatment for compound elements in the current approach. Exactly half of the defined thirty-two-element set defined by GEMINI permits multiple occurrences; little direct attempt has been made to address these (*Topic category*, *Subject* and *Extent* aside). A number of compound elements can be catered for using supplemental content templates and defaults as needed; entities otherwise populated will most likely necessitate manual intervention.

The most apparent advantage of the approach is the number of metadata items produced, although further benefits may also be seen. For one, metadata content is more dependent upon the data depicted, increasing the likelihood that dataset characteristics are more accurately reflected in the surrogates produced. The process that facilitates this admittedly only displaces rather than eliminates the burden of effort from data documentation to data management, and could thus be construed as representing a false economy. However, the contention is that the increased emphasis on dataset configuration and categorisation practices will invariably have positive consequences for data asset quality and management, while simultaneously supporting eventual metadata output.

Discussion and Conclusions

What has been presented can be viewed as a holistic approach to metadata automation. Various data and metadata creation, update and management process steps have been decomposed and subsequently employed to contribute towards the automated generation of geospatial metadata records. Standardised data encoding and storage practices have been extended beyond their traditional use in facilitating data upkeep, location and query to provide tangible support to metadata authoring. Entities drawn from both a dataset's enveloping computing environment and its internal constituents contribute towards documentation efforts through a system of annotation that permits such objects to be consistently referenced via a custom utility. What remains is a limited number of elements that require manual processing prior to human mediated quality control.

There nevertheless remains scope to further extend the current work. Demands for incorporating the socio-political context in which datasets are created and curated (as suggested by Comber *et al.* (2007)) could be met by harvesting or linking to 'project-level' metadata such as that maintained within the Scottish Executive's GI Projects Index Registry (part of their 'One Scotland – One Geography' initiative). Treatment for experiential metadata suggested by the same study could further be accommodated via the inclusion of voice recognition technology within the host GIS to enable easy capture of user-centric data perspectives while it is being manipulated and in turn enhance metadata richness. Similar provision could be included within surveying equipment to permit registering of erstwhile neglected information such as the environmental conditions at the instance of data capture, providing for more detailed data quality assessment at the feature level if later required.

Complete abstract entries could be approximated through the use of pre-composed sentence templates into which seed values may be inserted; whether this would act to discourage human input or complicate validation should be considered. Instantiating a reference boundary dataset library as an accessible Application Programming Interface (API) -driven web service meanwhile would avoid the need to bundle such layers within a custom utility and offer a single point of management when updates to boundary data are required. Further scope for exploiting dataset geometry may also exist through the development and application of pattern matching approaches; a catalogue of distinctive geographic features for instance could be used to extrapolate spatial reference details of data assets lacking such information through the cross-referencing of geographic footprints with those of known aspect. Potential for improvement may also exist where the underlying metadata standard is concerned. The *Frequency of update field* for instance is employed to indicate the prescribed data maintenance period; dynamically binding this element with data update events would provide a more accurate reflection of management history, rather than an arbitrary frequency estimate that may or may not be adhered to.

We would offer that the value of implementing a feature metadata schema goes farther than providing a more sophisticated means of data tracking. Indeed, the advent of feature-centric strategies such as multi-user corporate databases and Internet-enabled delivery mechanisms like WFS has witnessed an increase in the ability to remotely access the atomic components of data resources, rendering the need to retrieve host datasets in their entirety unnecessary. Associated efficiency gains (i.e. transport, processing) are nevertheless tempered by an uncertainty surrounding the pedigree of data thus retrieved, as conventional feature delivery approaches inconsistently depict the metadata of source collections. In contrast, the difficulty of collating resource documentation when deriving data from multiple-sourced feature sets illustrates the problem when dataset metadata is indeed accommodated. Implementing a metadata strategy focussed on the feature level may well be warranted in either case: whether to complement existing dataset metadata with more detailed information on individual features, or to contribute towards the automated generation of metadata subsequent to deriving new datasets from disparately sourced feature aggregations.

As is often the case with software deployment, the contribution of a particular solution will tend to be maximised when tailored to the local context, the consequent trade-off being solution portability. While the current work is presented within a specific context in terms of data configuration, metadata output and software application, it nevertheless alludes to the requirements a more generic approach not bound to specific domains or computing environments would need to satisfy. First, access to geospatial data stores should be transparent and not reliant upon the presence of specific, third party software. The personal geodatabase format adopted above was chosen so as to restrict the degrees of freedom of the current analysis and also due to it being a commonly used format positioned between single-user file-based hybrid storage strategies and multi-user, integrated stores. It is however a closed proprietary format and thus necessitates the use of its host application to provide full access to its constituents. Initiatives such as the open source Feature Data Objects (FDO) project[33] could provide significant assistance in bypassing such restrictions, providing an extensible API for "manipulating, defining, and analyzing geospatial information regardless of where it is stored." Second, attribute data should be amenable to analysis and mining. Access to the geometric content of datasets is to a degree predictable across different formats (cf. return bounding box, get projection); consistent polling of aspatial data on the other hand is approximated

[33] http://fdo.osgeo.org/. It should be noted that the personal geodatabase remains inaccessible via this and other non-vendor APIs.

through the imposition of standardised schema. Of the three strategies mentioned (keyword index of semi-structured data; formal spatial data standards; feature metadata), implementing an interoperable feature metadata approach may present the most promise, particularly in scenarios of deriving data from multiple sources as mentioned earlier.

Third, eventual metadata output should not be restricted to a particular standard or profile. The choice of convention has direct bearing on what elements are automatically generated and the format in which they are output. The emergence of ISO-based standards as the dominant initiative within the geospatial community arguably makes the task of catering for multiple output formats less problematic, permitting the elaboration of a base specification from which custom profiles may be derived. And finally, encapsulating a generic approach within a platform independent solution will maximise adoption and avoid marginalisation of any single user community. Existing initiative-driven metadata editors, whether browser-based or developed using a cross-platform software development kit (SDK) such as Sun Microsystem's Java SDK provide a sound basis on which more comprehensive metadata management approaches unhindered by operating environment-related restrictions can be extrapolated.

In the end, conventional metadata creation is unlikely to ever overcome its perception as an inconvenience, no matter how intensively its benefits are espoused. It therefore behoves the proponents of metadata practices to find ways of mitigating the burden of authorship, rather than solely pursuing the traditional dual-pronged 'carrot and stick' strategies that currently pervade. Regardless of whether a high-impact, customised approach is taken or whether a generic solution is developed for broader consumption, a niche for both certainly exists. Conditions such as the volume of available resources, incumbent computing infrastructure, in-house expertise and perhaps most pertinently, the extensiveness of geospatial data holdings will all come to bear on the choice of strategy providing the best fit for a given organisation. While implementation of any system designed to augment the computational support offered to metadata producers is far from trivial, we have demonstrated that the potential return on investment of effort can be considerable.

References

Agarwal P., 2005. Ontological considerations in GIScience. *International Journal of Geographical Information Science,* 19 (5), 501-536.

AGI 2004. *UK GEMINI Standard Version 1.0 - A Geo-spatial Metadata Interoperability Initiative* [online]. Available from: http://www.govtalk.gov.uk/schemasstandards/metadata_document.asp?docnum=903 [Accessed 29 March 2009].

Aloisio G., Milillo G. and Williams R.D., 1999. An XML architecture for high-performance web-based analysis of remote-sensing archives. *Future Generation Computer Systems,* 16, 91-100.

Balsoy O., Jin J., Aydin G., Pierce M. and Fox G., 2005. Automating metadata Web service deployment for problem solving environments. *Future Generation Computer Systems,* 21, 910-919.

Batcheller J.K., 2008. Automating geospatial metadata generation—An integrated data management and documentation approach. *Computers and Geosciences,* 34 (4), 387-398.

Beaujardière J. d L., Mitchell H., Raskin R. and Rao A., 2000. The NASA Digital Earth Testbed. In *ACM-GIS 2000, 8th ACM Symposium on Advances in Geographic*

Information Systems, 10-11 November 2000, Washington, D.C. New York, USA: ACM, 47-53.

Boucelma O. and Colonna F-M., 2005. Mediation for Online Geoservices. *In*: C Claramunt, Y-J Kwon and A Boujou eds. *Web and Wireless Geographical Information Systems: 4th International Workshop, W2GIS 2004, Lecture Notes in Computer Science 3428*. Berlin, Germany: Springer-Verlag, 81-93.

Boxall J., 2003. Geolibraries: geographers, librarians and spatial collaboration. *The Canadian Geographer,* 47 (1), 18-27.

Buttenfield B.P., 2002. Transmitting vector geospatial data across the Internet. *In:* M. Egenhofer, D. Mark eds. *Second International Conference on Geographic Information Science 2002, Lecture Notes in Computer Science 2478*. Berlin, Germany: Springer-Verlag, 51-64.

Comber, A.J., Fisher, P.F. and Wadsworth, R.A., 2007. User-focused metadata for spatial data, geographical information and data quality assessments. *In*: M. Wachowicz and L. Bodum, eds. *10th AGILE International Conference on Geographic Information Science*, 8-11th May 2007. Aalborg, Denmark: Aalborg University, p.p. 13.

Crompvoets, J., Bregt, A., Rajabifard, A. and Williamson, I., 2004. Assessing the worldwide developments of national spatial data clearinghouses. *International Journal of Geographical Information Science,* 18 (7), 665-689.

Davey, A. and Murray, K., 1996. Update on the National Geospatial Database - Collaboration between Organisations. *In Proceedings of the 8th Conference of the Association for Geographic Information at GIS96*, 24-26 September 1996. Birmingham,UK. London, UK: AGI, 1-6.

Department of the Environment 1987. *Handling Geographic Information*. The report of the Committee of Enquiry chaired by Lord Chorley. London, UK: Her Majesty's Stationary Office.

Devillers R., Gervais M., Bédard Y. and Jeansouin R., 2002. Spatial Data Quality: from Metadata to Quality Indicators and Contextual End-User Manual. In *OEEPE/ISPRS Joint Workshop on Spatial Data Quality Management*, 21–22 March, Istanbul, Turkey, 45-55.

Devogele T., Parent C., and Spaccapietra S., 1998. On spatial database integration. *International Journal of Geographical Information Science,* 12 (4), 335-352.

Dunfey, R. I., Gittings, B. M. and Batcheller, J. K., 2006. Towards an Open Architecture Vector GIS. *Computers and GeoSciences,* 32 (10), 1720-1732.

Fonseca F.T. and Egenhofer M.J. 1999. Ontology-driven Geographic Information Systems. In *ACM-GIS 1999, 7th ACM International Symposium on Advances in Geographic Information Systems*, 2-6 November, Kansas City. New York, USA:ACM, 14-19.

Frank A.U. 1998. Metamodels for Data Quality Description. *In:* MF Goodchild, R Jeansoulin eds. *Data Quality in Geographic Information – From Error to Uncertainty*. Paris, France:Hermes, 15-29.

Göbel, S. and Lutze, K., 1998. Development of meta databases for geospatial data in the WWW. *In 6th International Symposium on Advances in Geographic Information Systems, ACM-GIS '98*, 6-7 November 1998 Washington D.C., USA. New York, USA: ACM, 94-99.

Greenberg, J., 2004. Metadata Extraction and Harvesting: A Comparrison of Two Automatic Metadata Generation Applications. *Journal of Internet Cataloging,* 6 (4), 59-82.

Greenberg, J., Spurgin, K. and Crystal, A., 2006. Functionalities for automatic metadata generation applications: a survey of metadata experts' opinions. *International Journal of Metadata, Semantics and Ontologies,* 1 (1), 3-20.

Guptill, S. G., 1999. Metadata and data catalogues. *In:* P. Longley, M. F. Goodchild, D. J. Maguire, and D. W. Rhind, eds. *Geographical Information Systems*. Chichester, UK: Wiley, 677-692.

Healey R.G. and Delve J., 2007. Integrating GIS and data warehousing in a Web environment: A case study of the US 1880 Census. *International Journal of Geographical Information Science,* 21 (6), 603-624.

Higgins C., Medyckyj-Scott D. and Reid J. 2003. A Community Specific SDI - the Case of UK Academia. In *Geodaten- und Geodienste-Infrastrukturen - von der Forschung zur praktischen Anwendung*, June 26th-27th, Münster. Münster, Germany: University of Münster, 77-89.

Hill L.L., Janée G., Dolin R., Frew J. and Larsgaard M., 1999. Collection Metadata Solutions for Digital Library Applications. *Journal of the American Society for Information Science,* 50 (13), 1169-1181.

Hunter G.J. 2001. Spatial data quality revisited. *In Proceedings of GeoInfo 2001*, Rio de Janeiro, Brazil, 1–7.

ISO, 2005a. BS EN ISO 19115:2005. Geographic Information - Metadata. Failand, Bristol, UK: BSi British Standards.

ISO, 2005b. BS EN ISO 19109:2005 Geographic Information - Rules for Application Schema, Failand, Bristol, UK: BSi British Standards.

Kiehle C., 2006. Business logic for geoprocessing of distributed geodata. *Computers and Geosciences,* 32: 1746-1757.

Klein E. and Lutz M., 2005. The Role of Spatial Relations in Automating the Semantic Annotation of Geodata. *In*: A.G. Cohn, D.M. Mark eds. *Conference on Spatial Information Theory (COSIT) 2005, Lecture Notes in Computer Science 3693*. Berlin, Germany: Springer-Verlag, 133-148.

Kokla M. and Kavouras M. 2001. Fusion of top-level and geographical domain ontologies based on context formation and complementarity. *International Journal of Geographical Information Science,* 15 (7), 679-687.

Lutz M. and Klien E., 2006. Ontology-based retrieval of geographic information. *International Journal of Geographical Information Science,* 20 (3), 233-260.

Maguire D.J. and Longley P.A., 2005. The emergence of geoportals and their role in spatial data infrastructures. *Computers, Environment and Urban Systems,* 29, 3-14.

Mathys, T., 2004. The Go-Geo! Portal Metadata Initiatives. *In*: A Lovett, ed. *Proceedings of the Geographical Information Science Research UK Conference*

(GISRUK) 12th Annual Conference, 28-30th April 2004. Norwich, UK: University of East Anglia, 148-154.

Nanson, B., Smith, N. and Davey, A., 1995. What is the British National Geospatial Database? *In 7th Conference of the Association for Geographic Information*, 22 November 1995, Birmingham, UK. London, UK: AGI, 1-8.

Nebert, D.D., 2004. *Developing Spatial Data Infrastructures: The Spatial Data Infrastructure Cookbook v2.0* [online]. Available from: http://www.gsdi.org/gsdicookbookindex.asp [Accessed 29 March 2009].

Nogueras-Iso, J., Zarazaga-Soria, F.J., Béjar, R. and Álvarez, P.J. 2005. OGC Catalog Services: a key element for the development of Spatial Data Infrastructures. *Computers and Geosciences,* 31, 199-209.

Nogueras-Iso, J., Zarazaga-Soria, F.J., Lacasta, J., Béjar, R. and Muro-Medrano P.R., 2004. Metadata standard interoperability: application in the geographic information domain. *Computers, Environment and Urban Systems,* 28, 611-634.

OGC, 2003. OpenGIS Reference Model (OGC 03-040). Wayland, MA: OGC, p.p. 17.

Peng Z-R., 2005. A proposed framework for feature-level geospatial data sharing: a case study for transportation network data *International Journal of Geographical Information Science,* 19 (4), 459-481.

Timpf S., Raubal M. and Kuhn W., 1996. Experiences with Metadata. *In: Proceedings of the 7th International Symposium on Spatial Data Handling (SDH '96)*, 12-16 August 1996, Delft. Delft, The Netherlands: Delft University of Technology, 31-43.

Tsou, M.-H., 2002. An Operational Metadata Framework for Searching, Indexing, and Retrieving Distributed Geographic Information Services on the Internet. In M. Egenhofer and D. Mark eds. *Geographic Information Science (GIScience 2002), Lecture Notes in Computer Science Vol. 2478*. Berlin, Germany: Springer-Verlag, 313-332.

Tulloch D.L. and Robinson M., 2000. A progress report on a U.S. National Survey of Geospatial Framework Data. *Journal of Government Information,* 27, 285-298.

Turner A., 2006. *Introduction to Neogeography* [online]. O'Reilly Press. Available from:
http://www.oreilly.com/catalog/neogeography
[Accessed 29 March 2009]

Westbrooks E.L., 2004. Distributing and Synchronizing Heterogenous Metadata for the Management of Geospatial Information Repositories for Access. *In:* D Hillmann and E Westbrooks eds. *Metadata in Practice: Building the Diverse Digital Library*. Chicago, USA: American Library Association, 139-157.

Xie, R. and Shibasaki, R., 2007. Imagery Metadata Development Based on ISO/TC 211 Standards. *Data Science Journal,* 6, 28-45.

Yang, B., Purves, R., and Weibel, R., 2007. Efficient transmission of vector data over the Internet. *International Journal of Geographical Information Science,* 21 (2), 215-237.

Implementing feature-level semantics for spatial data discovery: supporting the reuse of legacy data using open source components

Research Paper V

Contributor: Femke Reitsma

Abstract

As the diversity and volume of web-based spatial resources continue to grow, so too do the challenges posed for those wishing to locate spatial information on the web, or for those wishing to contribute to it. Alongside the wealth of existing data holdings, differing conceptual models are leading to the increasing adoption of methods emerging from Semantic Web initiatives. This paper describes an approach that employs such methods with the aim of supporting the discovery and use of spatial resources, presented within the context of a search application. Based upon an architecture of freely available open source software components, the approach incorporates formal semantics through the use of an application ontology, associated with an RDF representation of natively-held geospatial data. The demonstrated prototype is offered as the first step towards a strategy for countering semantic ambiguity in data and query content, all the while providing a viable path for publishing legacy spatial data holdings on a future Geospatial Semantic Web.

Introduction

Real world abstractions, such as those intended for Geographical Information Systems (GIS), are invariably influenced by the context in which they are made. Personal and organisational perspectives, together with a varied and ever-evolving surveying and modelling techniques have all contributed to a diversity in how our immediate environment and beyond is viewed, recorded and analysed (Comber *et al.* 2007). Evidence of this within the geospatial community can be seen across the multiplicity of file formats, storage strategies and schema used to structure data (Comber *et al.* 2004, Nikolaos *et al.* 2005, Batcheller *et al.* 2007a). The consequences of such variability have undoubtedly been complicated by the shift from monolithic, single-user, domain specific approaches towards distributed, multi-user paradigms, as characterised by the evolution of GI (Geographic Information) systems. This tendency towards network-enabling software, services and data has led them to be less likely viewed in isolation (Kuhn 2005); current information systems are being superseded by those open to inter-communication, whether via Application Programming Interfaces (API) or otherwise accessible code libraries. A corollary has been a significant increase in the demand and availability of geospatial data and the services that exploit them. All have combined to highlight both the importance of interoperability initiatives and the appeal of automating formerly manual processes (Nikolaos *et al.* 2005, Greenberg *et al.* 2006).

Established information management systems are more often seen as insufficient when confronted with such challenges, leading to the increasing adoption of

Semantic Web-influenced initiatives (Herman 2007), as advocated by Berners-Lee (Berners-Lee *et al.* 2001). Representing a fundamental departure in how information is administered, a crucial aspect lies in how it is encoded, described, accessed and processed in a way more aligned with an increasingly distributed and interconnected computing landscape. Efforts to this end have largely focussed on the World Wide Web Consortium's (W3C) specifications for RDF (Resource Description Language) and ontology dialects such as RDF Schema (RDFS) and OWL (Web Ontology Language) (Passin 2004). Offering formal mechanisms for modelling data and their semantics in a way that allows automated processing, such specifications and the concepts that underpin them have formed the basis of much research in the domain of information management and knowledge representation. This has been especially true within the GIS community. A small subset of the raft of geo-centric research includes: geospatial query precision and recall, and data interoperability (Fonseca and Egenhofer 1999, Cruz *et al.* 2002, Fonseca *et al.* 2002, Nogueras-Iso *et al.* 2004); extending traditional metadata approaches for geographic datasets (Pfoser *et al.* 2002, Schuurman and Leszczynski 2006); geospatial data retrieval and service brokering (Visser *et al.* 2002, Lutz and Klien 2006, Wiegand and García 2007); and geospatial data and service discovery (Klien *et al.* 2006, Lutz and Kolas 2007).

This paper aims to present an approach to express and use feature level semantics, which is implemented to support spatial data discovery, with a particular emphasis on its value for legacy data sets. This work is distinct from the aforementioned research for geospatial data discovery and use as it focuses on semantics of the feature rather than the dataset level. The architecture is built upon open source components and driven by a geospatial dataset maintained in its native form. Designed in the context of a local search application, it is intended to represent the first step towards a method for publishing legacy geospatial data holdings on what has become known as the Geospatial Semantic Web (Fonseca and Rodriguez, 2007). As such, the approach presents an option for employing formal semantics in a fashion that does not require data transformation and enables continued exploitation of a geo-centric database system. The paper is structured accordingly: after a description of a number of standards-based initiatives and how conventional information management systems may succumb to problems of content ambiguity, we provide an overview of RDF and an application-focussed ontology, how they may be employed to capture formal data semantics and describe a given domain of discourse. This is followed by an outline of the architecture and the role of the contributing components presented in the context of sample use cases. Finally a critique of the approach is provided and conclusions reached.

Background

Geospatial standards

Prevailing standard-based methods for instantiating and manipulating Geographic Information (GI) have largely been motivated by interoperability concerns. The Open Geospatial Consortium's (OGC) Geography Mark-up Language (GML) and the International Organisation for Standardisation's (ISO) 19139 schema implementation provide examples of such methods, and highlight the popularity of eXtensible Mark-up Language (XML) techniques for serialisation of geospatial data and metadata. While these and similar standards have undoubtedly contributed towards geospatial data dissemination initiatives, it has been argued that the increasingly distributed nature of geospatial systems serves to highlight a number of their shortcomings (Probst *et al.* 2004, Klien *et al.* 2006). The GML specification, for example, enables a syntactically interoperable way of encoding geospatial data in a non-proprietary manner that is suited for online exchange. Nevertheless, it provides no formal mechanism for specifying semantic content. For example, feature types intended as synonymous, such as road and streets in differing GIS layers, will be handled as distinct entities. Similarly, ISO 19139 allows the structuring of metadata documents in a predictable fashion but offers little by way of incorporating what such structures signify[34]. The benefits afforded by interchangeable, human-legible XML thus start to recede when moving from a familiar domain to an unfamiliar one; syntax may be readily transformed mechanically but there is no certainty that its meaning will be interpreted as intended.

Content ambiguity - the Internet

In a context not particular to the geospatial domain, query precision and recall of web-based search engines also illustrate content ambiguity issues arising from a lack of treatment for semantics (Antoniou and van Harmelen 2004). Internet content and location is indexed and stored for subsequent retrieval but in a fashion that typically pays little heed to its inherent meaning. Consequently, pinpointing appropriate

[34] ISO 19139 is admittedly intended for use in conjunction with the content standard ISO 19115, but the point nevertheless holds true – it provides no treatment for semantics on its own.

information from a large, indiscriminate result set can be a difficult and time-consuming process; on the other hand, the term-sensitive nature of query mechanisms can result in fulfilling semantically related requests inconsistently. Long-established approaches driven by relational database management systems (RDBMS) are prone to similar shortcomings. Semantically related terms are retrieved only when they are explicitly queried for against the database, an especially unrealistic proposition for web-facing data stores where the end-user's detailed foreknowledge of its contents is improbable. For instance, a query to find all roads and streets will require that both search items be specified; querying on either will return an incomplete result set (assuming their interchangeable use).

Content ambiguity – the geospatial domain

The aforementioned trend towards adopting multi-user data storage strategies may well exacerbate content ambiguity in the context of the geospatial domain. Datasets arising from different sources in single-user format may often encode their attribute content in a semantically mismatched fashion, whether by design due to diverging protocols, or unintentionally as a result of differing perspectives; while frequently unwelcome, such discordances are arguably expected if not monitored for (Figure 1(a)). In more collaborative systems the scope for introducing inconsistent content is enhanced, particularly in the absence of rigorous protocols (Figure 1(b)).

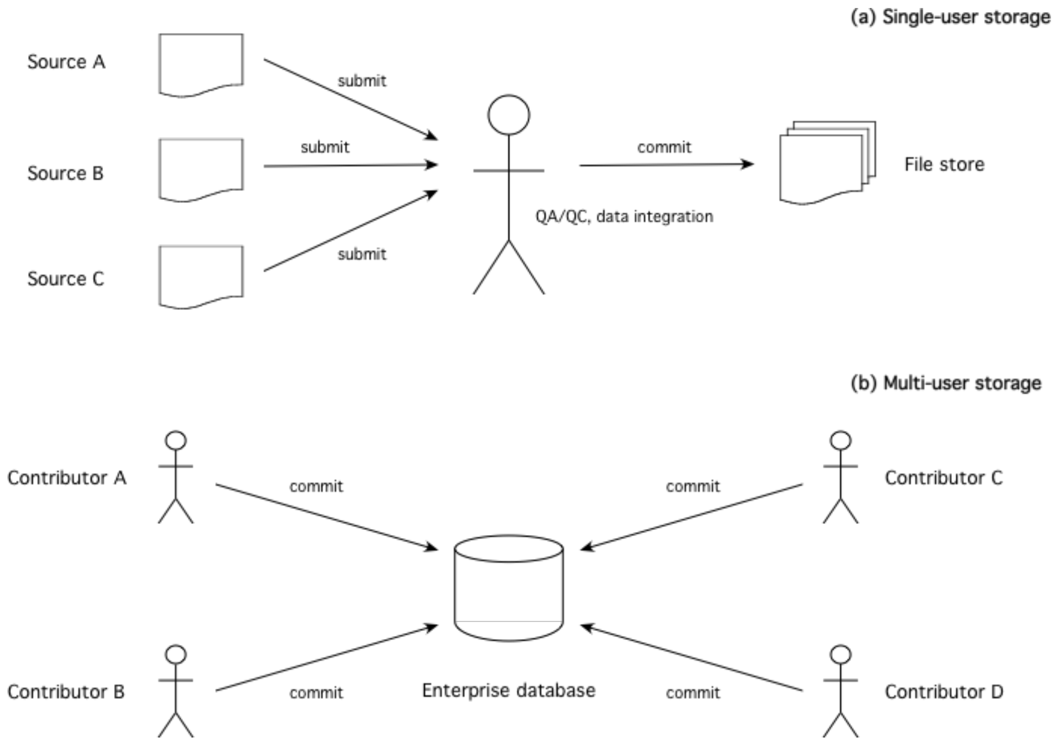


Figure 1(a) Integration of file-based data from multiple sources should ensure quality control and assurance (QA/QC) procedures are in place; a single perspective (data custodian), one potential point of failure. **(b)** Multiple contributors to an enterprise database with no intermediate integration or QA/QC in place prior to data commits; several perspectives, several potential points of failure.

While such complications feed the argument for standardised approaches, such standards are not unquestionably embraced (Guptill 1999, West-Jr. and Hess 2002), a perhaps justifiable stance when the perceived effort in achieving compliance overshadows the advantages that accompany it (Batcheller 2008). Conversely, the very use of traditional specifications can exacerbate semantic ambiguity where they are applied in an inconsistent or ill-conceived manner. Limiting compliance efforts to the schema level will facilitate dataset transformation but fail to address concerns relating to the dataset's content, standards for which may not be enforced whether due to oversight or verbosity (e.g. applying an ISO 19109 application schema using value type definitions and value domains derived from ISO 19115 (Batcheller *et al.*, 2007b)). In the case of legacy data holdings, they may simply be too cumbersome or incumbent procedures too entrenched, rendering whole-scale data migration initiatives unrealistic.

Approach Overview

It is in light of the foregoing that the opportunities afforded by Semantic Web approaches have been explored with the objective of enhancing the discovery and exploitation of geographic information (Nikolaos *et al.* 2005). Accordingly, a method for incorporating formalised semantics for marking-up geospatial attribute data using RDF and an associated OWL-extended RDFS domain ontology is presented, representing a terminological approach (Egenhofer 2002). The prototype developed therefore mitigates the consequences of content ambiguity in attribute data and facilitates inference over its contents via the axiomatic structure of the accompanying OWL vocabulary and the concepts defined within it.

The application was developed as a local, web-based search service using open source software components. Providing information on the location of buildings offering commercial services in an urban environment, the presented use-case is intended to demonstrate an everyday relevance while recognising the rise in popularity of local, geographically oriented search (Himmelstein 2005, Delboni *et al.* 2007). Instance data depicting building polygons and their attributes are held within an integrated database server, and a buildings ontology (Figure 3.) serves to describe the modelled domain. In contrast to approaches that extract and process geographic content held within unstructured web documents (Delboni *et al.* 2007), an underlying geospatial data model is employed, allowing direct exploitation of geometric constructs for location-based query fulfilment.

A further consequence of maintaining data in a geospatial format is that the data need not be replicated within an external document, a departure from other approaches employing RDF data surrogates as distinct file-based serialisations (Jenkins *et al.* 1999). As the volume and magnitude of existent databases increase, the effort and cost of semantically enabling data through the use of such surrogates renders their creation increasingly unrealistic. If legacy data is to be exploited based upon their semantics there is a growing need for it to be accessed natively (Bizer and Seaborne 2004). Accordingly, the prototype retains all instance data in its native state from where it is virtualised as RDF on demand. Layers can thus be updated without any need to take them off-line; undesired artefacts arising from asset conversion such as data loss (Piwowar *et al.* 1990, Fileto 2001) and update latency associated with maintaining asset surrogates (Badard and Richard 2001) are also largely circumvented.

Semantic Framework

Resource Description Framework

RDF provides a standard data model in which information is represented as statements about resources, where the meaning of these statements describing resources and the relationships among resources is formally defined in an ontology.

Each statement models a data resource, a property describing the resource's behaviour and the property's value, whether in the form of a literal or another resource. These *subject*, *predicate*, *object* triples (Figure 2) are uniquely identified by Uniform Resource Identifier (URI) references, the values of which may indicate temporary constructs such as parameter variables or locations of actual network resources. Deploying data in RDF thus enables its representation in an unambiguous, semantically rich format suited to machine-driven processing, exchange and reuse. Applied in the context of a geospatial data layer, RDF can provide a feature-level metadata strategy facilitating a semantically explicit description of attribute table content and consequently individual features, that would otherwise exist implicitly in a database schema definition or associated database column. At the analogous dataset level, collections of RDF statements (visually represented as graphs) may be expressed in a number of serialisation formats, the most common being XML and Notation 3 (N3).

```
<http://eeo.ed.ac.uk/bldno364>
       < http://eeo.ed.ac.uk/edbuildings.owl#hasBusinessName>
              <"Kebab Majhal">
```

Figure 2. RDF triple represented as an N3 statement and illustrated by a directed graph. Subject / object node pairs are connected by an arc representing the statement's predicate.

Ontology

The application ontology was constructed using the Protégé 3.1 editor and instantiated in OWL-Full, a machine-readable knowledge representation language grounded in first order logic. Whereas RDF allows the association of meaning with atomic units of data within a specific domain, OWL extends this notion by providing a means of semantically modelling the domain as a whole through the definition of (potentially hierarchical) concepts and the relationships that hold between them. Built upon RDF and typically serialised in RDF/XML, OWL details a range of RDFS and language-specific constructs that are essentially map-able to Description Logic (DL) modelling primitives that facilitates automated reasoning over encoded instance data.

Concepts are expressed within the ontology as OWL classes, used to categorise collections of similar individuals in what is also referred to as the class's extension. Organised into a class hierarchy, each superclass is specialised by (or subsumes) one or more subclasses to form a taxonomy (or ontology subset) modelling the objects of the domain of discourse (Figure 3).

Figure 3. Graphical representation of the application's Building ontology.

Atomic semantic descriptions may be assigned to each class and combine to form axioms that define class behaviour. As illustrated in Figure 4, a Building class axiom is constructed using a combination of the subclass construct (`rdfs:subClassOf`), a restriction (`owl:Restriction`), a union collection (`owl:unionOf`) and a value constraint (`owl:someValuesFrom`). Assigning such properties to class instances allow incorporation of further semantic content within the ontology, and by extension, the instance data to which it is applied.

```
<owl:Class rdf:ID="Building">
    <rdfs:subClassOf rdf:resource="http://www.w3.org/2002/07/owl#Thing"/>
    <rdfs:subClassOf>
      <owl:Restriction>
        <owl:someValuesFrom>
          <owl:Class>
            <owl:unionOf rdf:parseType="Collection">
              <owl:Class rdf:ID="CommercialBuilding"/>
              <owl:Class rdf:ID="PublicBuilding"/>
              <owl:Class rdf:ID="PrivateBuilding"/>
            </owl:unionOf>
          </owl:Class>
        </owl:someValuesFrom>
    </owl:Restriction>
    </rdfs:subClassOf>
  </owl:Class>
```

Figure 4. RDF/XML Building class axiom and its three immediate subclasses CommercialBuilding, PublicBuilding and PrivateBuilding.

One implication of using OWL to classify data for the purposes of providing reasoning support is the need to define explicit relationships lest inappropriate conclusions (statements) are reached (inferred) (such as membership of multiple classes when only one is intended). While this can serve to complicate ontology generation, such specification undoubtedly helps to further enrich the semantic description of the data modelled and assists in its validation. Considering the Building class axiom in Figure 4 above, the CommercialBuilding, PublicBuilding and PrivateBuilding may be further defined as disjoint i.e. building individuals may only belong to one of the classes. Used in combination with `owl:unionOf` (an AND operator), complete membership of the superclass is specified.

The property `owl:complementOf` meanwhile permits the definition of a binary NOT relationship between classes, describing a class whose individuals are not members of the complement class. In the current local search application this is implemented for members of the Restaurant class; VegetarianRestaurant is defined as the complement of OmnivorousRestaurant (Figure 5) and thus permits negation-based queries such as “return all restaurants in area X that are not vegetarian”.

```
<owl:Class rdf:ID="OmnivorousRestaurant">
  <owl:Class>
    <owl:complementOf>
      <owl:Class rdf:ID="VegetarianRestaurant"/>
    </owl:complementOf>
   </owl:Class>
   <rdfs:subClassOf>
     <owl:Class rdf:ID="Restaurant"/>
   </rdfs:subClassOf>
</owl:Class>
```

Figure 5. RDF/XML fragment illustrating the use of the `owl:complementOf` construct.

Differing data perspectives may be accommodated using the OWL property `owl:sameAs`, allowing for the use of two distinct URI references to denote the same identity (i.e. intensional meaning) for a given resource. Consider the scenario of a collaborative data creation project designed to record places of interest in a manner such as the Open Street Map project. Following a recent visit to the British Isles, Contributor A, of North American origin submits details of her favourite drinking venues as “Bar”; Surveyor B, from the UK encodes them as “Pub”. Conventional queries against the database based on one or the other are mutually exclusive despite their assumed shared identity. The `owl:sameAs` property may be used to link occurrences of the two. By mapping the class Bar to the class Pub, all subsequent queries against the data store aimed at retrieving instances of Bar will return any instances encoded as Pub.

Custom properties introduce further semantic content and permit inclusion of formerly implicit resource characteristics. Consider the arguably trivial yet illustrative case of locating a suitable venue for an after-work tipple. Operations run against the PublicHouse class, individuals of which can be defined as establishments selling alcohol, will retrieve all and only instances beneath this taxonomic root.

Identification of establishments categorised elsewhere in the ontology that would otherwise fulfil the 'serves alcohol' criterion are precluded, such as occurrences of student unions and cinemas with accompanying bar facilities. Associating a custom datatype property with all individuals exhibiting this behaviour (via a `hasAlcoholLicence` property) can address this problem; the property is so assigned to the PublicHouse class via its class axiom. Other complying instances can be individually assigned this property within the instance data and hence will be 'seen' in subsequent property-based retrieval operations.

Properties may also be organised hierarchically, permitting their specialisation of sub-properties in a fashion similar to that for classes above via the `rdfs:subPropertyOf` axiom. Following the above theme, `hasAlcoholLicence` can be 'subclassed' by the `servesBeer`, `servesWine`, `servesCocktails` properties, allowing the inheritance of the parent property's behaviour while facilitating more detailed data annotation, query and reasoning support.

RDF Toolkit

Once the domain of discourse has been modelled, a means of interacting with the ontology and instance data is necessary. The Jena semantic toolkit[35] provides one such option, comprising of a library of Java classes complete with a series of APIs through which RDF graphs and vocabularies can be managed and manipulated.

[35] http://jena.sourceforge.net/

Graphs sourced locally or across the web may be loaded into memory as Jena models where they are queried, reasoned over and potentially output as new graphs.

Inference support is provided through a number of internal Jena reasoners that facilitate machine 'interpretation' of RDF graphs and their ontologies. These inference engines 'reason' largely on the basis of first order logic, implementing the axioms of the specific ontology language with which they are associated. Such operations produce logical extrapolations from the instance data and ontology as additional RDF assertions that are combined with said inputs to form an inference or entailment model. In other words, the entailment model not only contains the base assertions present in the source data but also further virtual statements derived through its binding with the ontology and reasoner; implicit associations within the data are thus entailed and made available for query. Using the taxonomy illustrated in Figure 3, Bar is asserted to be a subclass of PublicHouse, which is in turn asserted as a subclass of CommercialBuilding. Given the transitive nature of the `rdfs:subClassOf` property, the inference engine deduces that Bar is also a subclass of CommercialBuilding and hence includes the assertion in the entailment model (Figure 6).

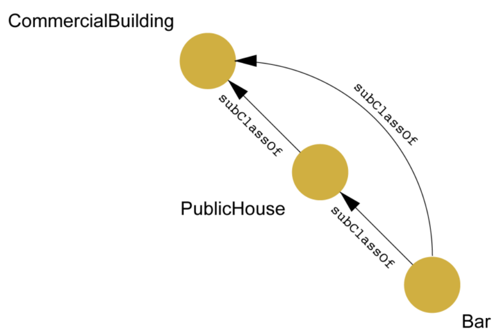


Figure 6. The transitive `rdfs:subClassOf` property as implemented in the application ontology.

Extracting information from base assertion graphs and the entailment models derived from them is supported using the RDF query language SPARQL[36]. Implemented using the ARQ query engine for Jena, SPARQL fulfils a role analogous to that of SQL within relational database systems. It functions by matching basic subject, predicate, object (s,p,o) patterns specified in a WHERE clause against the triples of the model, returning information in the form of bound variables or RDF subgraphs. Figure 7 illustrates using a solitary N3 statement as data and the SPARQL query necessary to determine the name of the occupying business.

```
Data:

<http://eeo.ed.ac.uk/bldno364> <jkb:hasBusinessName> "Kebab Majhal" .

Query:
SELECT ?business WHERE {
        <http://eeo.ed.ac.uk/bldno364> <jkb:hasBusinessName> ?business .
        }

Result:
Kebab Majhal
```

Figure 7. RDF triple and a simple SPARQL query to extract the RDF object based on given subject and predicate. The predicate's prefix (jkb) indicates the application's ontology namespace, i.e. http://eeo.ed.ac.uk/jkb/edbuildings.owl#

Data Store

Instance data is stored within a PostGIS-extended PostgreSQL object-relational database, adopted as it provides a widely supported, license free, potentially scalable option complete with support for geometric constructs. PostgreSQL offers a platform independent option complete with native programming interfaces; PostGIS in effect geospatially enables the PostgreSQL database server by facilitating the incorporation of the OGC's Simple Features for SQL specification and providing support for

[36] A recursive acronym for *SPARQL Protocol and RDF Query Language*, a World Wide Web Consortium (W3C) Recommendation.

common spatial operators. The configuration can act as a back-end to a variety of desktop and web-based GIS applications, and thus offers a suitable representation of a native geospatial data store for the current purpose. Polygon data for a city centre was attributed in ArcGIS and loaded into the database server via the shapefile interchange format where it could be subsequently visualised and manipulated in situ using the open source Quantum GIS client. Within the dataset, individuals described by the ontology's Buildings class are represented by features corresponding to individual database rows; a datatype property is drawn from a field denoting a building name attribute, if known. Buildings of a commercial purpose have further fields populated to represent its primary category of business, business name, and whether it is licensed.

Database field	**Ontology construct**	**Construct type**
building_id	Building	Class
structname	hasBuildingName	Property
str_use_id	BusinessType	Class
bus_name	hasBusinessName	Property
license	hasAlcoholLicense	Property

Table 1. Database attribute fields and their mapping to ontology constructs.

RDF virtualisation

Once a data store has been prepared there are two main routes to manipulating its contents as RDF: transforming it via an export text file or interacting with it in situ. The Jena framework provides an interface for accessing persistent database content as RDF, but only if that content is stored in triple form. For the current database schema the D2RQ[37] platform circumvents any such requirement, extending Jena and enabling an RDF-centric view of the attribute content by representing it as a virtual graph. In so doing D2RQ negates the need for the creation of data snapshots and enables interaction with live database content natively. The D2RQ Mapping Language describes the correlation between the ontology and the PostgreSQL/PostGIS data, detailed in a mapping file (Figure 8). Automatically generated using the accompanying generate-mapping script and tailored for the application, the RDF/XML file contains ClassMaps that define how class instances are identified within the database; PropertyBridges are similarly used to associate properties with such instances. The D2RQ Engine uses this map to generate a virtual, read-only RDF model of the registered dataset; it also handles Jena API calls, translating them into database-native SQL queries and returning the results in RDF.

[37] http://www4.wiwiss.fu-berlin.de/bizer/d2rq/

```
@prefix map: <http://eeo.ed.ac.uk/jkb/ed_map.n3#> .

@prefix jkb: <http://eeo.ed.ac.uk/jkb/edbuildings.owl#> .
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
@prefix xsd: <http://www.w3.org/2001/XMLSchema#> .
@prefix d2rq: <http://www.wiwiss.fu-berlin.de/suhl/bizer/D2RQ/0.1#>
.

map:database a d2rq:Database;
      d2rq:jdbcDriver "org.postgresql.Driver";
      d2rq:jdbcDSN "jdbc:postgresql://eeo.ed.ac.uk:5432/pg_db_sp";
      d2rq:username "pguser";
      d2rq:password "pgpwd";
.

# Table edinburgh
map:edinburgh a d2rq:ClassMap;
    d2rq:dataStorage map:database;
    d2rq:uriPattern
"http://eeo.ed.ac.uk/bldno@@edinburgh.building_id@@";
      d2rq:class jkb:Building;
.

map:BusinessClassMap a d2rq:ClassMap;
    d2rq:uriPattern "bldno@@edinburgh.building_id@@";
    d2rq:dataStorage map:database;
 .

map:BusinessType a d2rq:PropertyBridge;
    d2rq:property rdf:type;
    d2rq:uriPattern
"http://localhost/edbuildings.owl#@@edinburgh.str_use_id@@";
    d2rq:belongsToClassMap map:BusinessClassMap;
```

(a)

(b)

(c)

(d)

Figure 8. A subset of the D2RQ map file. (a): the list of namespaces used; (b): the D2RQ database map defining the connection type (JDBC) and connection parameters; (c): the root Building class map associating each database row with an instance of Building; (d): the Business class map and accompanying property bridge that associates each value in the str_use_id column with the appropriate instance of the BusinessType class.

Application Walkthrough

User interaction takes place via a map and form driven web page. Basic geometric parameters such as the centre of an area of interest or its enveloping bounding box are selected via a web-mapping interface; attribute parameters such as buffer distance and subject-predicate-object patterns are be specified in standard HTML form objects. Server-side code accepts submitted inputs, calls application logic and returns results from the underlying data store, presented in map form. An overview of the application's architecture is presented in Figure 9 and its major components discussed below.

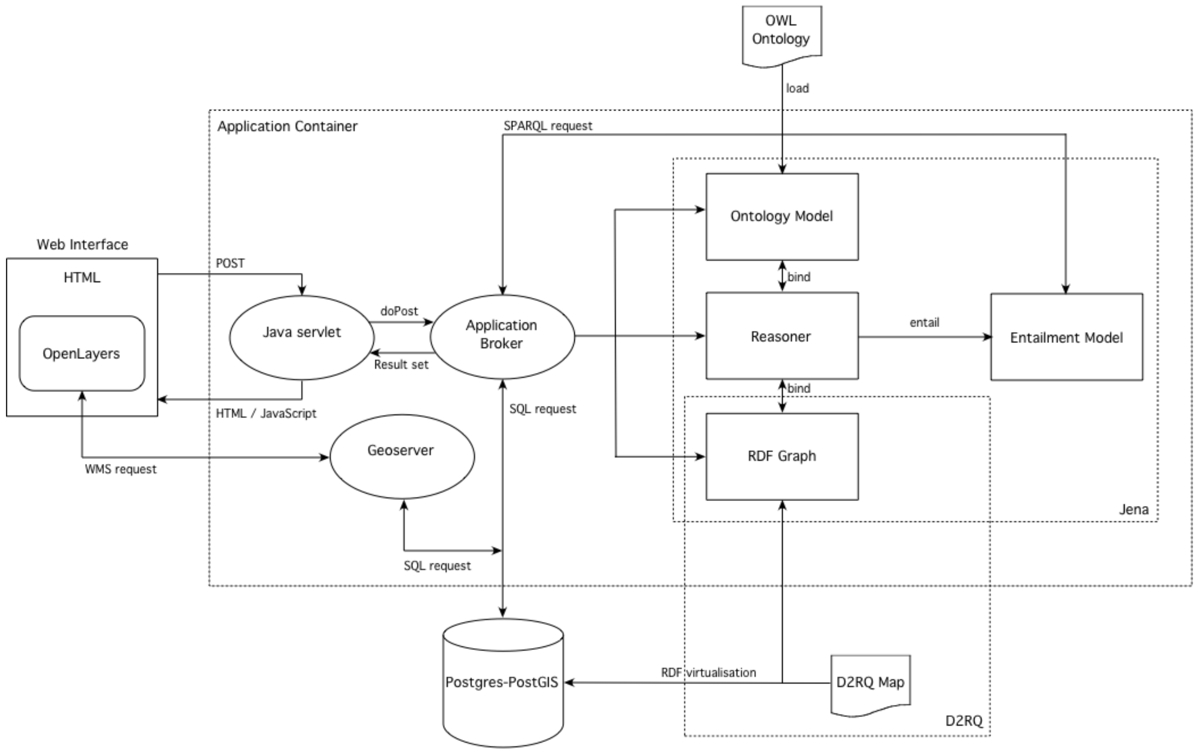


Figure 9. Prototype architecture overview. User interaction is browser-based; variables chosen are submitted to the application logic, deployed within an Apache Tomcat application container. Attribute queries are executed against the entailment model using SPARQL; spatial queries are executed directly against the database server using SQL. WMS requests such as GetMap (rendering) and GetFeatureInfo (feature identification) are handled by the Geoserver instance.

Map Interface

The map interface was implemented using the OpenLayers JavaScript library embedded in an HTML page (Figures 10, 11). A standard OGC Web Mapping Service (WMS) provided by the open source Geoserver serves Postgres/PostGIS data for map rendering. Mouse position is converted to dataset coordinates on the fly; geographic features can be queried via on-click events that initiate the WMS's GetFeatureInfo API. Geometric parameters used within the application logic are captured using further JavaScript events and are parsed and passed to invisible text fields within the same form designed to accept user-defined subject-predicate-object (s,p,o) patterns; on submit the browser initiates a POST request to the Apache Tomcat web container in which the application logic is deployed.

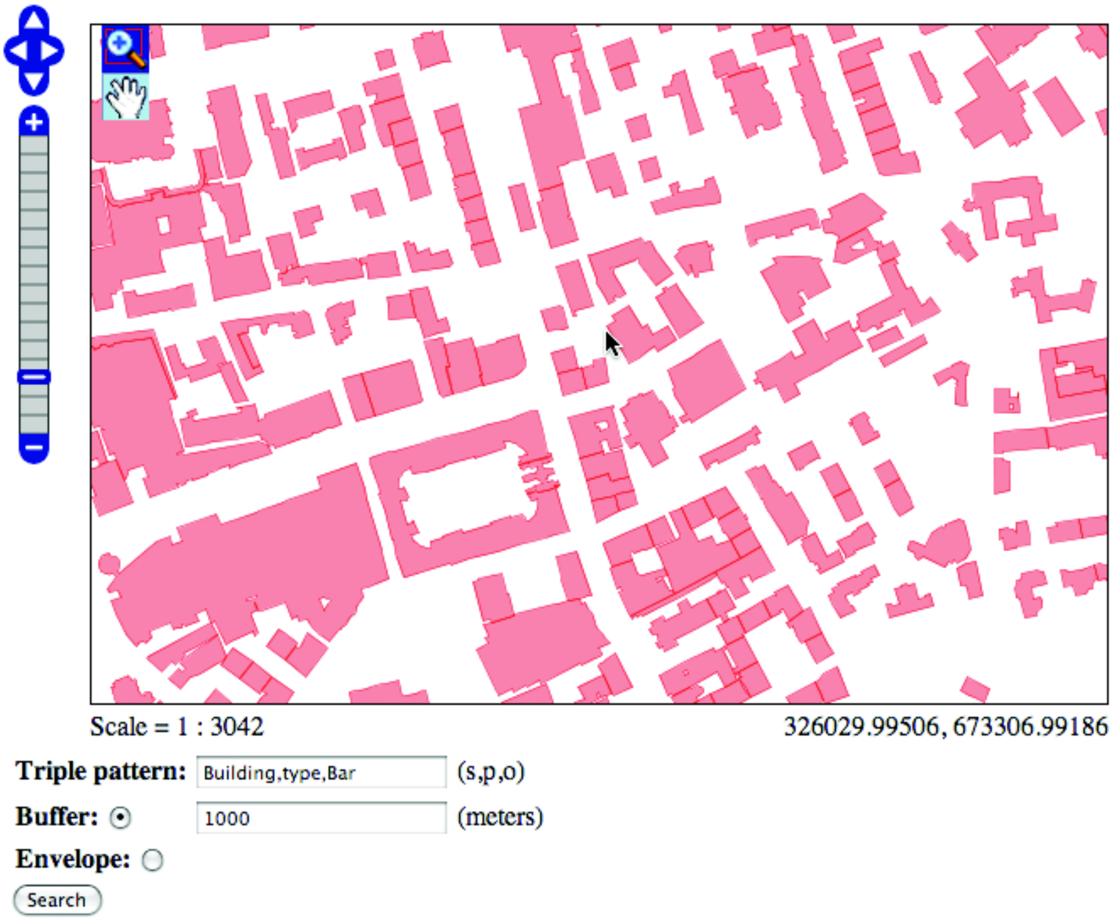


Figure 10. The application user interface. A point of interest is selected; all features matching the specified pattern that occur within the specified buffer are returned. The sample query will identify all buildings whose primary commercial function is categorised as either ‘Bar’ or ‘Pub’ within a 1km distance of the designated point.

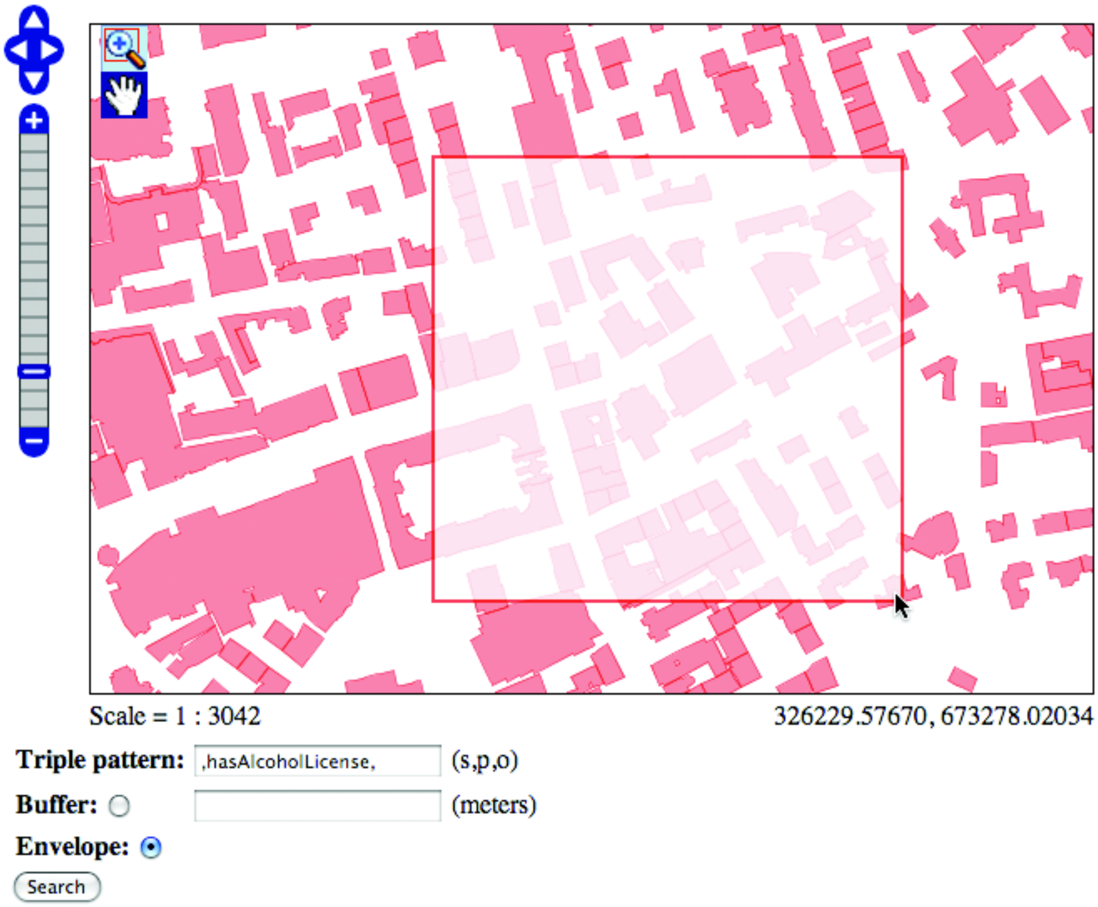


Figure 11. The application user interface. A query envelope is selected; all features matching the user's pattern within it are returned. The sample pattern will return any feature (building) within the envelope that has an associated hasAlcoholLicense property.

Name-value pairs corresponding to the form fields are passed to a Java servlet configured to accept incoming requests through a pre-determined port. Within the servlet a doPost method commences invocation of the application logic. Helper classes are employed to instantiate point and triple objects from the accepted parameters that are subsequently passed, along with buffer distance or extent, to an application broker (Figure 12).

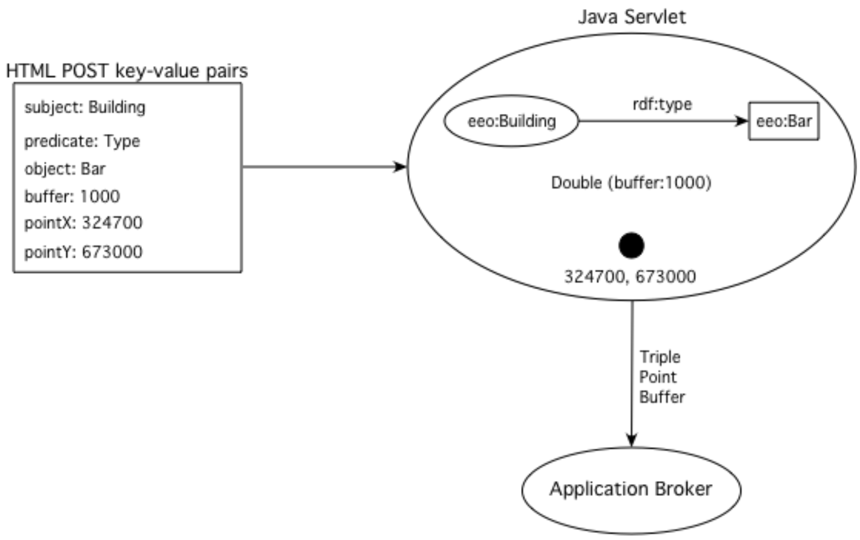


Figure 12. Initial role of the application's servlet, transforming HTML POST key-value pairs from form text. Triple and Point objects are instantiated and passed to the application's broker along with the casted buffer variable (when used).

Instantiating RDF objects

Once instantiated, the application broker loads the ontology into memory as a Jena model where it is coupled with an instance of the inbuilt OWL-mini reasoner[38]. Using the D2RQ map, the D2RQ Engine retrieves instance data via a JDBC connection and transforms it into RDF triples on the basis of the file's ClassMaps and PropertyBridges. Another Jena model is formed as a result and again bound to the reasoner to form the entailment model; the broker constructs the necessary SPARQL query from the submitted pattern and executes it against the model (Figure 13(a)).

[38] Despite using OWL-Full to encode the application ontology, an OWLMini reasoner was employed due to Jena's incomplete 'out-of-the-box' support for OWL-Full, as well as for better prototype performance.

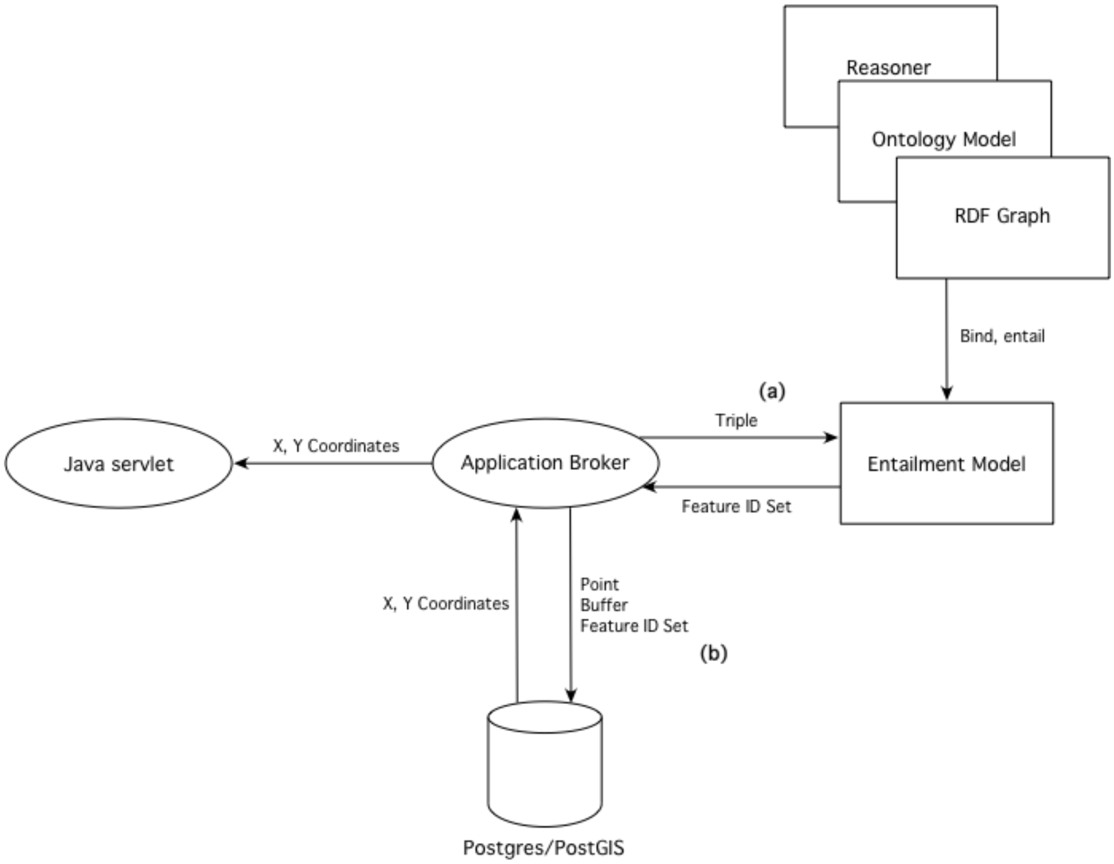


Figure 13. The application's broker logic matches triple patters against the entailment model using SPARQL (a). Feature identifiers are extracted and combined with input point and buffer (or envelope points alone) to form a spatial SQL statement, executed against the database server (b). X,Y coordinate pairs are returned to the Java servlet for inclusion within the result map.

Querying the PostGIS layer

Upon retrieval of a non-empty SPARQL result set, feature identifiers in the result set are extracted from and used along with the specified spatial and attribute conditions to generate a SQL statement to run against the geospatial data. Another JDBC connection is opened to the database through which the coordinates of all features matching the query are returned (Figure 12(b). The results are returned to the main Java servlet as a collection of point objects, terminating the application broker instance.

Rendering results

Once the point set has been accepted, the servlet begins generating the final HTML output. X and Y coordinates are parsed, cast to the appropriate datatype and passed via a JavaScript array to an embedded OpenLayers map where the results are presented for inspection (Figure 14).

Figure 14. The rendered result map with a user-interrogated point feature.

Discussion and Conclusions

What has been presented represents a proof of concept for a means of applying semantic-based techniques to geospatial data description and interrogation. Informed by established Semantic Web / W3C approaches and implemented using free and open source software (FOSS) and common programming methods, an architecture has been developed that offers a first attempt at addressing content ambiguity and facilitating semantic reasoning of geographic data held in its native form at the feature level. As the use of file-based data surrogates is not required, complications frequently associated with data conversion and update latency are largely circumvented. And as it is based upon a multi-user data store, geographic layers can be maintained live and in a solitary location.

A consequence of presenting a proof of concept is that the degrees of freedom in which it is presented must by necessity be confined – scope for improving upon and extending the present work most certainly exists. Specifying attribute data in the form of triple subject-predicate-object patterns, while adequate for current demonstrative purposes, would clearly not suffice in a publicly-deployed solution. Such queries may be relatively straightforward to formalise for the developer but do not provide an intuitive mechanism for the average user (Mäkelä *et al.* 2006). Future revisions of the current application should be informed by ongoing research in the field of Semantic Web user interaction (e.g. Catarci *et al.* 2003, Athanasis *et al.* 2004).

The next logical step for the application may well be to expose it as a machine-consumable service, as is another aspect of the (Geo-) Semantic Web vision. To this end, the D2R Server[39] tool could be used to extend the D2RQ Engine to enable access by external web applications to the virtualised RDF database, again via the SPARQL protocol. With the current Java servlet-based approach in mind, one

[39] http://www4.wiwiss.fu-berlin.de/bizer/d2r-server/

possible approach would be to refactor the servlet using the Apache Axis2/Java web services engine, thus wrapping the D2R Server - Postgres/PostGIS configuration for deployment as a RPC[40] or REST[41] service. While the argument for which service type to adopt is subject to a much wider debate (Richardson and Ruby 2007, p.299), a preliminary review suggested that a RESTful approach may be more in keeping with the philosophy of the Semantic Web (Lathem *et al.* 2007, Battle and Benson, 2008).

Exposing legacy data as RDF adds a further layer to the application stack, and hence degrades performance in comparison to purely database-driven approaches. While this may be an acceptable trade-off for applications of smaller scope, the appropriateness of the current configuration must be questioned for more substantial data holdings. Preliminary indications point to the instantiation of the entailment model and its interrogation as the primary performance bottleneck; furthermore, the implication of deploying alternative reasoning engines provides room for added investigation. Given that the development focus was on initial concept implementation and not on fine-tuning however, the role of the elaborated logic and its reliance upon a language occasionally criticised for its performance (Java) should not be dismissed. Indeed, robust conclusions are difficult to substantiate in the absence of a rigorous performance analysis. Load testing would also be required to determine exactly how scalable the architecture is; of related interest would be to

[40] Remote Procedure Call
[41] REpresentational State Transfer

investigate how the architecture could be migrated to popular corporate data management solutions such as those based upon Oracle or ESRI's ArcSDE.

One implication of the approach taken is that no provision for geospatial reasoning was made - outside of those geometric operations supported by PostGIS. While the open nature of the adopted data store did not in itself preclude access to geographic coordinate data for the purposes of implementing such reasoning support, the lack of a known, robust geospatial reasoner at the time of writing resulted in a focus on the attribute component. Further, it has been argued that while geospatial reasoning is a ever-evolving field of research, spatial data constructs are not yet accommodated within current Semantic Web languages such as OWL, nor in the logics that underpin them (Reitsma and Hiramatsu 2006). Considering however the open nature of the presented architecture and the provision within the Jena framework for incorporating external reasoning engines via its implementation of the DIG[42] interface, the possibility of implementing spatial reasoning in future application iterations exists.

A corollary and potential criticism of the current approach is the need for two distinct querying mechanisms for spatial and aspatial data. Indeed, it may be seen to reflect early hybrid GIS data models (c.f. ESRI's coverage) since foreshadowed by integrated strategies generally considered more robust due to their having solitary query brokers for interacting with the data held. While there is the potential in the current configuration for spatial-attribute mismatches in the event of update being

[42] Description Logic Implementation Group

committed between the separate queries, the risks are considered negligible as RDF virtualisation is executed ‘on-the-fly’. Further, data are only served to the web user in this manner; integrity of the underlying layer is preserved due to its maintenance by the integrated Postgres/PostGIS database, whether through the SQL command line or via a GIS client.

A final observation regarding the architecture’s hybrid approach to querying spatial and attribute data pertains to its potential applicability to proprietary data strategies. Formats such as ESRI’s personal geodatabase obscure their spatial components within Binary Long Data type fields that are not openly accessible outside the Arc platform; attribute data may however be accessed using the open source software MDB Tools (Dunfey *et al.* 2006). In such instances a hybrid approach similar to that outlined herein becomes necessary if reasoning support is to be sought. Considering the increasing purchase semantic technologies have gained within the mainstream data management industry however[43], viable integrated geo-centric solutions may well be on the horizon.

With respect to the elaborated ontology, its application focus leaves it open to the critique voiced by Smith and Mark (2001) pertaining to what constitutes a ‘good’ ontology. While partially informed by a published vocabulary[44], the present ontology

[43] Oracle have introduced OWL inferencing and RDF support in their most recent iteration of their flagship database product 11g: http://www.oracle.com/technology/tech/semantic_technologies/index.html, WWW page retrieved January 29 2008

[44] http://frot.org/space/0.1/

was specifically tailored to meet the initial requirements of the current proof of concept. As a consequence, it does not currently make the allowances for a wider depiction of reality (Smith and Mark 2001). While such a consideration is not dismissed, the aim was to demonstrate a pragmatic solution to implementing a semantically enabled system within a specific context. As Dean (2007) observed, however, ontologies facilitate mark-up in a manner that does not modify the data depicted; replacing the incumbent vocabulary with one more 'aware' of the wider world remains an option in future work. In any case, consideration of the broader discussion surrounding such inclusive ontological initiatives serves as a reminder that Semantic Web technologies do not offer a panacea for issues of interoperability. Challenges still exist not only if conflicting perspectives with regards to data representation and interpretation are to be accommodated, but also to cater for the varying levels of granularity and breadth of scope used to frame abstractions of the real world, as dictated by the context in which they are made.

Finally, this paper has focussed on dealing with legacy data, yet there is an opportunity to reconsider how we capture and represent the semantics of spatial data in general. Our traditional geospatial data model involves associating with a point or region in space and time some measurable quality, such as temperature, or observable feature, such as a building, as has been recently reviewed by Goodchild *et al.* (2007). When capturing data we implicitly subscribe to some kind of conceptualisation. If we can make this explicit in an ontology we can extend our fundamental representation of geospatial data to take advantage of our conceptualisation and the potential for reasoning with formal semantics. By including a URI in our basic data model that links it to our ontology defining our conceptualisation, we gain the power of formal semantics to reason with the concepts represented in our spatial data sets. For example, we could extend Goodchild *et al.*'s geo-atom (2007) with the addition of a URI: <x, Z, z(x), URI>. In future we might have ontologies associated with our measurement instruments such that they automatically associate their ontology with the data captured, providing a kind of feature level metadata that can be reasoned with as per the example provided in this paper.

References

Agarwal, P., 2005. Ontological considerations in GIScience. *International Journal of Geographical Information Science*, 19 (5), 501-536.

Antoniou, G. and van Harmelen, F., 2004. *A Semantic Web Primer*. London, UK: MIT Press.

Athanasis, N., Christophides, V. and Kotzinos, D., 2004. Generating on the fly queries for the semantic web: The ICS-FORTH graphical RQL interface (GRQL). *In:* S.A. McIlraith, D. Plexousakis and F. van Harmelen eds. *Proceedings of the 3rd International Semantic Web Conference (ISWC2004), Hiroshima, Japan, Lecture Notes in Computer Science 3298*. Berlin: Springer, 486–501.

Badard, T. and Richard, D., 2001. Using XML for the exchange of updating information between geographical information systems. *Computers, Environment and Urban Systems*, 25, 17-31.

Batcheller, J.K., 2008. Automating geospatial metadata generation—An integrated data management and documentation approach. *Computers and Geosciences*, 34 (4), 387-398.

Batcheller, J.K., Gittings, B.M. and Dowers, S., 2007a. The Performance of Vector Oriented Data Storage Structures in ESRI's ArcGIS. *Transactions in GIS*, 11 (1), 47-65.

Batcheller, J.K., Reitsma, F. and Gittings, B., 2007. Implementing ISO-compliant Feature Metadata. *In*: A.C. Winstanley, ed. *Proceedings of the Geographical Information Science Research UK Conference (GISRUK) 15th Annual Conference*, 11-13th April 2007. Maynooth, Republic of Ireland: National University of Ireland Maynooth, 72-78.

Battle, R. and Benson, E., 2008. Bridging the semantic Web and Web 2.0 with Representational State Transfer (REST). *Web Semantics: Science, Services and Agents on the World Wide Web*, 6 (1), 61-69.

Berners-Lee, T., Hendler, J. and Lassila, O., 2001. The Semantic Web. *Scientific American,* 284 (5), 34-43.

Bizer, C. and Seaborne, A., 2004. D2RQ - Treating Non-RDF Databases as Virtual RDF Graphs. *In Proceedings of the 3rd International Sematic Web Conference (ISWC2004)*, November 7-11 2004, Hiroshima, Japan, 2.

Catarci, T., Di Mascio, T., Franconi, E., Santucci, G. and Tessaris, S., 2003. An Ontology Based Visual Tool for Query Formulation Support. *In: Proceedings of the*

OTM 2003Workshops: On The Move to Meaningful Internet Systems 2003, Lecture Notes in Computer Science 2889. Berlin, Germany: Springer, 32-33.

Comber, A., Fisher, P. and Wadsworth, R., 2004. Integrating land-cover data with different ontologies: identifying change from inconsistency. *International Journal of Geographic Information Sciences,* 18 (7), 691-708.

Comber, A.J., Fisher, P.F. and Wadsworth, R.A., 2007. User-focused metadata for spatial data, geographical information and data quality assessments. *In*: M. Wachowicz and L. Bodum, eds. *10th AGILE International Conference on Geographic Information Science*, 8-11th May 2007. Aalborg, Denmark: Aalborg University, p.p. 13.

Cruz, I.F., Rajendran, A., Sunna, W. and Wiegand, N., 2002. Handling Semantic Heterogeneities Using Declarative Agreements. *In: Proceedings of the Tenth ACM International Symposium on Advances in Geographic Information Systems*, November 8-9, 2002, McLean, VA. New York, USA: ACM,168-174.

Dean, D.J., 2007. Characterizing Spatial Databases via their Derivation: A Complement to Content Ontologies. *Transactions in GIS,* 11 (3), 399-412.

Delboni, T.M., Borges, K.A.V., Laender, A.H.F. and Davis Jr, C.A., 2007. Semantic Expansion of Geographic Web Queries Based on Natural Language Positioning Expressions. *Transactions in GIS*, 11 (3), 377-397.

Dunfey, R.I., Gittings, B.M. and Batcheller, J.K., 2006. Towards an Open Architecture Vector GIS. *Computers and GeoSciences*, 32 (10), 1720-1732.

Egenhofer, M.J., 2002. Toward the Semantic Geospatial Web. *In: Proceedings of the Tenth ACM International Symposium on Advances in Geographic Information Systems*, November 8-9, 2002, McLean, VA. New York, USA: ACM, 1-4.

Fileto, R., 2001. Issues on Interoperability and Integration of Heterogeneous Geographical Data. *In: Proceedings of the 3rd Brazilian Symposium on Geoinformatics (GEOINFO2001)*, 4-5 October 2001, Rio de Janeiro, Brazil, 8.

Fonseca, F.T. and Egenhofer, M.J., 1999. Ontology-driven Geographic Information Systems. *In: Proceedings of the 7th ACM international symposium on Advances in geographic information systems, ACM-GIS 1999*, 2-6 November 1999, Kansas City, MO. New York, USA: ACM,14-19.

Fonseca, F. T., Egenhofer, M. J., Agouris, P. and Câmara, G., 2002. Using ontologies for integrated geographic information systems. *Transactions in GIS*, 6 (3), 231-257.

Fonseca, F. T. and Rodriguez, A., 2007. From Geo-Pragmatics to Derivation Ontologies: New Directions for the GeoSpatial Semantic Web. *Transactions in GIS*, 11 (3), 313-316.

Frank, A.U., 2003. A linguistically justified proposal for a spatio-temporal ontology. *In: Proceedings of the Workshop on Fundamental Issues in Spatial and Geographic Ontologies*, 23 September 2003, Leeds, UK.

Goodchild, M.F., Yuan, M. and Cova, T.J., 2007. Towards a general theory of geographic representation in GIS. *International Journal of Geographical Information Science,* 21 (3), 239-260.

Greenberg, J., Spurgin, K. and Crystal, A., 2006. Functionalities for automatic metadata generation applications: a survey of metadata experts' opinions. *International Journal of Metadata, Semantics and Ontologies,* 1 (1), 3-20.

Guarino, N., 1998. Formal Ontology and Information Systems. *In: Proceedings of 1st International Conference on Formal Ontology in Information Systems (FOIS'98)*, Trento, Italy 6-8 June 1998. Amsterdam, The Netherlands: IOS Press, 3-15.

Guptill, S. G., 1999. Metadata and data catalogues. *In:* P. Longley, M. F. Goodchild, D. J. Maguire, and D. W. Rhind, eds. *Geographical Information Systems*. Chichester, UK: Wiley, 677-692.

Herman, I., 2007. Semantic Web Adoption. *In: Proceedings of the First China Semantic Web Symposium (CSWS2007)*, 19 November 2007, Beijing, China p.p.45.

Himmelstein, M., 2005. Local search: The internet is the yellow pages. *IEEE Computer,* 38 (2), 26-35.

Jenkins, C., Jackson, M., Burden, P. and Wallis, J., 1999. Automatic RDF metadata generation for resource discovery. *Computer Networks,* 31, 1305-1320.

Klien, E., Lutz, M. and Kuhn, W., 2006. Ontology-based discovery of geographic information services—An application in disaster management. *Computers, Environment and Urban Systems,* 30 (1), 102-123.

Kuhn, W., 2001. Ontologies in support of activities in geographical space. *International Journal of Geographic Information Science,* 15 (7), 613-633.

Kuhn, W., 2005. Geospatial Semantics: Why, of What, and How? *In:* S. Spaccapietra, E. Zimányi eds. *Journal on Data Semantics III, Lecture Notes in Computer Science 3534*. Berlin, Germany: Springer, 1-24.

Lathem, J., Gomadam, K. and Sheth, A.P., 2007. SA-REST and (S)mashups : Adding Semantics to RESTful Services. *In: International Conference on Semantic Computing (ICSC 2007)*, 17-19 September 2007, Irvine, California. Washington D.C., USA: IEEE, 469-476.

Lutz, M. and Klien, E., 2006. Ontology-based retrieval of geographic information. *International Journal of Geographical Information Science,* 20 (3), 233-260.

Lutz, M. and Kolas, D., 2007. Rule-Based Discovery in Spatial Data Infrastructure. *Transactions in GIS,* 11 (3), 317-336.

Mäkelä, E., Hyvönen, E. and Saarela, S., 2006. Ontogator — A Semantic View-Based Search Engine Service for Web Applications. *In: Proceedings of the 5th International Sematic Web Conference (ISWC 2006), Athens, GA, USA Lecture Notes in Computer Science 4273.* Berlin, Germany: Springer, 847-860.

Nikolaos, A., Kostas, K., Michail, V. and Nikolaos, S., 2005. The Emerge of Semantic Geoportals. *In: Proceedings of the OTM Workshops: On The Move to Meaningful Internet Systems 2005, Lecture Notes in Computer Science Vol. 3762*. Berlin, Germany: Springer, 1127-1136.

Nogueras-Iso, J., Zarazaga-Soria, F.J., Lacasta, J. and Muro-Medrano, B.R., 2004. Metadata standard interoperability: application in the geographic information domain. *Computers, Environment and Urban Systems,* 28, 611-634.

Passin, T. B., 2004. Explorer's Guide to the Semantic Web. Greenwich, CT, USA: Manning.

Pfoser, D., Pitoura, E. and Tryona, N., 2002. Metadata Modeling in a Global Computing Environment. *In: Proceedings of the 10th ACM International Symposium* *on Advances in Geographic Information Systems (ACM-GIS2002)* McLean, VA. New York, USA: ACM 68-73.

Piwowar, J.M., Ledrew, E.F. and Dudycha, D.J., 1990. Integration of spatial data in vector and raster formats in a geographic information system environment. *International Journal of Geographic Information Science,* 4 (4), 429-444.

Probst, F., Ginotti, F., Morantes, A., Esbri, M., Barros, M., Gutierrez, M. and Kuhn, W., 2004. Connecting ISO and OGC Models to the Semantic Web. In: *Proceedings of the Third International Conference on Geographic Information Science,* Adelphi, MD, USA, p.p. 5.

Reitsma, F. and Hiramatsu, K., 2006. Exploring GeoMarkup on the Semantic Web. In: J. Suarez and B Markus eds. *Proceedings of the 9th International AGILE Conference on Geographic Information Science,*Visegrád, 20-22 April, 2006. Visegrád, Hungary: Unversity of West Hungary, 110-118.

Richardson, L. and Ruby, S., 2007. RESTful Web Services. Sebastopol, USA: O'Reilly Media, Inc.

Schuurman, N. and Leszczynski, A., 2006. Ontology-based metadata. *Transactions in GIS,* 10 (5), 709-726.

Smith, B. and Mark, D.M., 2001. Geographical categories: an ontological investigation. *International Journal of Geographic Information Science,* 15 (7), 591-612.

Visser, U., Stuckenschmidt, H., Schuster, G. and Vögele, T., 2002. Ontologies for geographic information processing. *Computers and Geosciences,* 28 (1), 103-117.

West-Jr., L. A. and Hess, T. J., 2002. Metadata as a knowledge management tool: supporting intelligent agent and end user access to spatial data. *Decision Support Systems,* 32, 247-264.

Wiegand, N. and García, C., 2007. A Task-Based Ontology Approach to Automate Geospatial Data Retrieval. *Transactions in GIS,* 11 (3), 355-376.

Discussion and Conclusions

This thesis has presented a number of ways of developing Geographical Information (GI) documentation practices. With the aim of better facilitating the ultimate utilisation of GI, the proposed approaches focused on the technological infrastructure supporting data discovery, the automation of documentation processes and the implications of describing geospatial information resources of varying granularity. Proprietary as well as free and open source software (FOSS) have been leveraged to provide both proof-of-concept and conceptual solutions, all of which have been detailed in the context of an existing body of literature. The research has resulted in the forerunning papers that explore GI documentation. Each paper successively contributes to the notion that geospatial resources are potentially better exploited when documentation practices account for the multi-granular aspects of GI, and the varying ways in which such documentation may be used.

Geospatial Metadata Hierarchy

Dataset (layer) metadata, as introduced in Paper I, have been seen to be instrumental in the utilisation of geospatial data, providing a means through which they may be published and discovered. They have been shown to facilitate the discrimination of candidate datasets, and provide details on how best to exploit the data they describe, their origin and provenance. They further allow data holdings to be tracked and may serve to institutionalise organisational memory for safeguarding data investments and future utility. While arguably the most prevalent category of geospatial metadata, dataset metadata represents only one tier in a possible geospatial metadata hierarchy (Figure 1). Extending metadata beyond, or within, the dataset to the feature-level as introduced in Paper IV, provided a way of registering quality and other information at a resolution that may otherwise remain neglected, and thus permitted a finer degree of data oversight for decision support and data management purposes. Further deconstructing sub-layer metadata into feature and attribute metadata (Paper V) has allowed for more precise access to the data they describe and further served to disambiguate between the value or meaning associated with features and attributes held within relational models, whether for eventual interpretation by humans or manipulation by machine. At the opposite scale, the project documentation of Paper II has been proposed as a preliminary reconnaissance mechanism to inform data acquisition efforts, provide visibility for initiatives within and out-with specific communities and offer a degree of insight into the context in which organisations utilise GI.

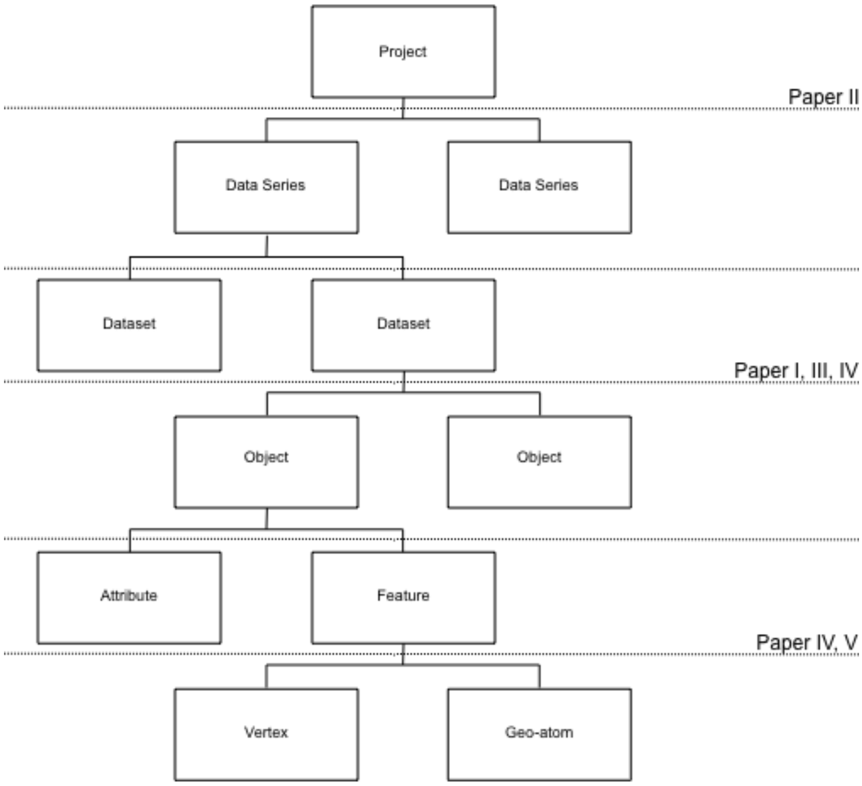


Figure 1. Geospatial metadata hierarchy and the papers that deal with the documentation of GI of specific granularity. Note that attribute and feature metadata comprise feature-level metadata.

These do not represent the only granularities of GI that may be usefully documented however. Data series metadata can be employed to describe attributes common to a collection of datasets, such as point, line and polygon data layers collected during a particular surveying mission, multiple imagery tiles flown on a certain date, or vector layers and the raster files over which they were digitised. Coupling datasets via series metadata can be of particular importance where features of distinct layers are associated temporally as well as spatially, such as survey points along a watercourse whose route deviates over time.

Geographic objects meanwhile occupy the conceptual space between the dataset and feature levels, representing aggregations of features within a layer. Some GIS software may cater for such entities through an object-relational model, allowing the association of properties or behaviours with objects, mirroring to some degree the functionality of object-oriented systems. Consider a land parcel boundary formed from a combination of man-made materials and natural features; for the purposes of managing material lifecycle replacement, capturing the boundary as a single polygon or multi-line feature may be undesirable. Registering each length of boundary individually allows for the persistence of feature-level information (e.g. material used per length, date scheduled for maintenance); using the object metaphor permits the grouping of constituent features, allowing for their collective manipulation and documentation (e.g. boundary use, date of boundary inspection).

Extending documentation to below the feature level may also be required in some circumstances. Vertices represent x and y coordinate pairs that define the shape line and polygon features assume; recording rudimentary metadata at this level may be warranted in data conflation scenarios where features of distinct provenance are merged into a solitary entity, or where features share a common vertex. At a similar scope is the geo-atom (Goodchild *et al.*, 2007), proposed as part of a simplified approach for representing geographic information. Based upon the supposition that all geographic representations are comprised of individual points that represent a geographic location x, a property of that location Z, and a value for the property, z(x), this theoretical data model raises the prospect of similar requirements for maintaining sub-feature metadata as with the vertex.

As shown in Paper IV, the effort placed in completing metadata at one level in the metadata hierarchy can be leveraged to contribute to that in another. In the presented context, sub-feature metadata would be aggregated and used to contribute to feature-level metadata, which would in turn be aggregated to contribute to object-level metadata, and so on, upwards along the metadata hierarchy (Figure 2). Conversely, lower tiers in the hierarchy can inherit metadata items registered at a higher tier when relevant, reusable content exists. It is unlikely however that a fully bi-directional exchange could occur, although the prospect of this could be investigated as a follow-on from the current study; where specific obstacles to data or information inheritance and aggregation occur, and the extent to which the processes could be automated. Calculating successive spatial extents from the geo-atom to project documentation for example presents one possible instance where metadata could traverse the entire hierarchy, from low tier to high; registering the geographic coverage at each granularity can facilitate spatial querying of each tier. Inheritance is a more restrictive proposition. The further up the hierarchy, the more abstract content typically becomes, by virtue of generalisation and the increasing 'distance' from the data being depicted. Narrative elements could conceivably be inherited from project documentation down to the dataset tier; beyond this the utility of potentially verbose items would be open to question.

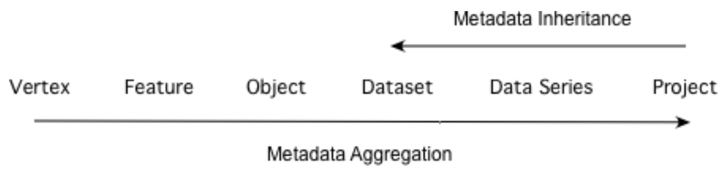


Figure 2. Metadata inheritance and aggregation. Aggregation may traverse the entire hierarchy; inheritance is likely to halt at the dataset level and not expected to occur beneath it.

Future Generation Geographical Information Infrastructures

Paper I dealt with traditional geospatial clearinghouses and their role in facilitating data discovery. Ancillary services, both bespoke and based upon OGC specifications were proposed to further advance these pioneering clearinghouses beyond a first generation capacity. While such advancement towards second generation SDI may serve to modernise and reinvigorate pioneering initiatives, they remain largely focussed upon the dataset paradigm. Access to data at the sub-dataset level is broached with the inclusion of Web Feature Services for asset visualisation, however features remain selected and rendered by virtue of their host dataset metadata. Even with the option of incorporating further web services to perform remote geoprocessing operations prior to data download, the fundamental model has persistently centred on data distribution. This clearly favours the GIS specialist, equipped with the tools and expertise necessary to extract information from the sorts of data commonly delivered via current SDI.

It has been against this backdrop that we have witnessed the emergence of contemporary “neogeography” techniques (Turner, 2006). Companies such as Google, Microsoft and Yahoo!, by entering the online mapping arena have opened the door to rudimentary GIS functionality for the general user, such as custom map making and route finding. For the web developer, the subsequent exposure of their proprietary logic via APIs has opened access to underlying map and data services for custom applications, fuelling the creation of geocentric flavours of Web 2.0 ‘mash-ups’, an approach characterised by the cherry picking of functionality from multiple online services and recombining them into novel, easy to use solutions. While such neogeography techniques have been viewed critically by some in the more traditional GIS community, the lessons that each can learn from the other should not be overlooked. As the arbitrary line between the two disciplines continues to blur, we are likely to see more by way of the quality indicators characteristic of geospatial metadata associated with data employed in mainstream web applications, with the probability of increasing rigour being placed in the quality control of the data consumed and output.

Conversely, initiatives such as GIgateway would undoubtedly benefit from the content ranking and referral functionality common to a number of Web 2.0 news aggregation sites, where participants rate content and develop reputations of trustworthiness based upon the merit of their contributions. Such feedback, or user-generated metadata, can provide an invaluable source of insight into the data being described, and hence better facilitate their evaluation. Similarly, SDI stand much to gain if the collaborative infrastructure and enthusiasm of participants seen in many online communities could be harnessed for data contribution, maintenance and documentation. Google Earth offers a case in point; users document locations around the globe (geo-annotation), upload photographs or other media content that incorporate coordinate information tying them to specific places (geo-tagging) and add their own data layers to a body of information available to the wider community. Within the professional GIS field, means of incentivising the continued and active participation in online geospatial data sharing initiatives continue to prove elusive. What a number of the aforementioned Web 2.0 sites illustrate is the willingness of those in a community to expend effort in adding value to a service that engages with their personal interests. Of course, identifying the common ground between what users find gratifying and what organisations contributing to SDI will benefit from will remain a difficult problem to solve.

The exertion of mutual influence on the techniques used by both the geospatial semantic web community and neogeographers is already evident however. A particular example has been the departure from traditional monolithic and client-server system architectures built upon closed code bases, to those increasingly capable of drawing upon and linking consumable services and data distributed across the World Wide Web – many using the same tools as introduced in Paper V. Together with research into formalised semantics for machine-interpretable documentation and the automated exploitation of these online resources, the prospect of future generation information frameworks such as Service-Driven Infrastructures (Craglia *et al.*, 2007) has begun to take shape. These propose to move past the data-centric remit of current SDI towards those geared to the provision of information; where average users can pose questions on portals driven by services with the ability to identify and chain appropriate data, processing and delivery resources, to ultimately return only the answers sought.

Many obstacles remain to be surmounted before any vision of a future generation SDI can be realised, but the core problem may be characterised by identifying how manual processes can be performed automatically. Natural language queries need to be interpreted and parsed; candidate services appropriate to the task at hand must be found and discriminated amongst; chosen services have to be combined in the right order to form a coherent workflow, and results must be distilled and delivered back in a manner suitable for the user. From a GI perspective, data will need to be not only discoverable and accessible to the automated routines to which they will be subjected to, but they will need to be unambiguously self-documenting, not least for selection and exploitation purposes. A multi-granular documentation strategy can potentially provide the necessary flexibility for responding to diverse service calls through the use of metadata appropriate to the data resolution required.

The combined geo-semantic web techniques employed in Paper V effectively represent a form of multi-granular documentation. The feature and attribute data comprising a dataset, documented using the RDF syntax of subject – predicate – object statements embody, for this application, its finest resolution of data. These statements combine to comprehensively describe each geographical feature, otherwise represented within the relational model as a database record or row. Through an associated ontology, classes and class behaviour are described for varying collections of statements and mapped to aggregations of data, allowing the definition of a nested hierarchy of geographic objects within the dataset. While the ontology further describes the behaviour of the dataset construct, it is sufficiently extensible to permit the description of further data layers, and thereby assume the role of data series metadata.

Parallels between these RDF and ontology-based techniques and the theory of the aforementioned geo-atom model may inform future work. The similarity of the subject–predicate–object triples of RDF and the location–property–value structure of the geo-atom in particular would suggest a natural fit for instantiating the model and would provide a scaleable means of data mark-up. Registering features via their constituent geo-atoms could be facilitated through the use of a point dataset. As with the approach employed for generic features in Paper V, each point feature (geo-atom) would denote an RDF subject, with property–value pairings mapped to the remaining RDF predicate and object components. In conventional GIS terms, the resulting graph of geo-atom–hasProperty–value statements would equate to a layer of single attributed points. With an associated ontology, geo-atoms would be aggregated into features and features into objects, as described above. Thus with some modification, the geo-semantic application of Paper V could be developed to provide a geo-atom based application for vector data. As the model elaborated by Goodchild *et al.* (2007) proposes to support continuous fields as well as discrete features, developing the application to incorporate raster data would present a further interesting challenge.

In its current incarnation, the geo-semantic application presents one approach for incorporating formal semantics within applications driven by data held within conventional GIS applications, bridging a gap between traditional geospatial and pioneering geo-semantic approaches. The passage of time is likely to witness developments that will continue to drive this convergence; the adoption of ontology and OWL technology within commercial products such as Oracle's 11g relational database management system offers one case in point. Faced with the prospect of increasingly complex software approaches however, an important challenge is ensuring the accessibility of newer technology. Even when techniques make the transition from theory and proof of concept to incorporation within mainstream solutions, their adoption within commonly employed toolsets is not assured. Theoretical and functional envelopes continue to be tested but the key is how these advances may be applied to common work practices, or transparently deployed in support of them.

Final Comments

The view has been that geospatial resource documentation, by organising and improving accessibility of GI, better facilitates the utilisation of the resources depicted. Despite the potential gains, the process of documentation and the investment it can demand leads to it remaining a low priority activity for many data producers and custodians, one that continues to be overlooked in favour of more immediate concerns. If documentation has been met with ambivalence by some, it has also been met with disdain from others. Incorrect, inconsistent or incomplete metadata-to-data mapping, subjective completion and update latency all have the effect of eroding confidence in both descriptions and that being described. The current work has presented just a few ways of mitigating some of these concerns while illustrating potential merit in multi-granular geospatial documentation.

Paper I demonstrated how the effectiveness of layer-level metadata can be improved by revisiting the metadata creation-publication lifecycle and identified ways in which the data – metadata divide can be mitigated to better enable data access for clearinghouse-based SDI. The methods proposed were presented in the context of the UK's public sector geospatial data sharing initiative GIgateway, and represented a unique – and arguably, long overdue – examination of the service. Paper III clearly showed that geospatial metadata automation is indeed feasible; Paper IV elaborated upon this and highlighted the parts geometry, and in particular, feature-level metadata can play in generating higher-order metadata. Paper II and the GI Projects Registry have demonstrated the utility of project documentation in raising the profile and awareness of GI-related activity while providing a service with the potential of better informing data pursuit operations. Finally, Paper V illustrated the fundamental problems that conventional syntactic approaches can be confronted by, and showed how feature-level semantic mark-up can be exploited to overcome such problems in the context of a geospatial web application. Combined, the papers establish the merit in documenting geospatial resources at varying granularities.

Nevertheless, instituting a comprehensive metadata strategy requiring data documentation at several granularities may well be excessive if not unrealistic for most mainstream geospatial applications. Extensive documentation of this kind would demand automation. Metadata items at different data resolutions would need to be accessible to both aggregation and inheritance operations, allowing the flow of fine-grained information in one direction and, where practical, the inheritance of courser information in the other. Pragmatically, the level of effort needed to coordinate and implement such a computationally driven metadata hierarchy in itself makes widespread adoption impractical. All the same, it is worth bearing in mind that as the volumes of geospatial data gathered continues to expand, so too the demand for ways of managing and maintaining the usefulness of potentially unwieldy data repositories. Considered and meticulous documentation can help meet this demand. A system of hierarchical metadata, of self-describing data, would provide for data retrieval unrestricted to a solitary granularity, improve the accessibility of GI and ultimately, its utilisation.

References

Craglia, M., Kanellopoulos, I. and Smits, P., 2007. Metadata: where we are now, and where we should be going. *In*: M. Wachowicz and L. Bodum, eds. *10th AGILE International Conference on Geographic Information Science*, 8-11th May 2007. Aalborg, Denmark: Aalborg University, p.p. 7.

Goodchild, M.F., Yuan, M. and Cova, T.J., 2007. Towards a general theory of geographic representation in GIS. *International Journal of Geographical Information Science,* 21 (3), 239-260.

Turner A., 2006. *Introduction to Neogeography* [online]. O'Reilly Press. Available from:
http://www.oreilly.com/catalog/neogeography
[Accessed 29 March 2009]

Appendices

Appendix A – Extended bibliography

Abdelmoty, A.I. and El-Geresy, B.A., 2004. Shema Visualisation using a Metadata Approach for GIS. *In*: A Lovett, ed. *Proceedings of the Geographical Information Science Research UK Conference (GISRUK) 12th Annual Conference*, 28-30th April 2004. Norwich, UK: University of East Anglia, 160-162.

Agarwal P., 2005. Ontological considerations in GIScience. *International Journal of Geographical Information Science,* 19 (5), 501-536.

AGI, 2007. Future Technology for a UK Metadata Service: Research Report. London, UK: AGI, p.p. 41.

AGI 2004. *UK GEMINI Standard Version 1.0 - A Geo-spatial Metadata Interoperability Initiative* [online]. Available from: http://www.govtalk.gov.uk/schemasstandards/metadata_document.asp?docnum=903 [Accessed 29 March 2009].

Aloisio G., Milillo G. and Williams R.D., 1999. An XML architecture for high-performance web-based analysis of remote-sensing archives. *Future Generation Computer Systems,* 16, 91-100.

Amin, S., 2003. The Open Archives Initiative Protocol for Metadata Harvesting: An Introduction. *In: DRTC Workshop on Digital Libraries: Theory and Practice*, 10-21st March 2003. Bangalore, India: DRTC, p.p. 13.

Anderson, G. and Moreno-Sanchez, R., 2003. Building Web-Based Spatial Information Solutions around Open Specifications and Open Source Software. *Transactions in GIS,* 7 (4), 447-466.

Anderson, J. D. and Pérez-Carballo, J., 2001. "The nature of indexing: how humans and machines analyze messages and texts for retrieval: part I: research, and the nature of human indexing." *Information Processing and Management: an International Journal,* 37 (2), 231-254.

Annoni, A. and Smits, P., 2003. Main problems in building European environmental spatial data. *International Journal of Remote Sensing,* 24 (20), 3887-3902.

Antoniou, G. and van Harmelen, F., 2004. *A Semantic Web Primer*. London, UK: MIT Press.

Arctur, D., Hair, D., Timson, G., Martin, E. and Fegas, R., 1998. Issues and prospects for the next generation of the spatial data transfer standard (SDTS). *International Journal of Geographic Information Science,* 12 (4), 403-425.

Athanasis, N., Christophides, V. and Kotzinos, D., 2004. Generating on the fly queries for the semantic web: The ICS-FORTH graphical RQL interface (GRQL). *In:* S.A. McIlraith, D. Plexousakis and F. van Harmelen eds. *Proceedings of the 3rd International Semantic Web Conference (ISWC2004), Hiroshima, Japan, Lecture Notes in Computer Science 3298*. Berlin: Springer, 486–501.

Badard, T. and Richard, D., 2001. Using XML for the exchange of updating information between geographical information systems. *Computers, Environment and Urban Systems,* 25, 17-31.

Balley, S., 2004, Interactive Specification of Customised Geographical Data Sets. *In*: A Lovett, ed. *Proceedings of the Geographical Information Science Research UK Conference (GISRUK) 12th Annual Conference*, 28-30th April 2004. Norwich, UK: University of East Anglia, 155-159.

Balsoy O., Jin J., Aydin G., Pierce M. and Fox G., 2005. Automating metadata Web service deployment for problem solving environments. *Future Generation Computer Systems,* 21, 910-919.

Batcheller J.K., 2008. Automating geospatial metadata generation—An integrated data management and documentation approach. *Computers and Geosciences,* 34 (4), 387-398.

Batcheller J.K., Dunfey R.I., Reitsma F.E. and Gittings B.M., 2007. Automating geospatial metadata using ESRI's ArcGIS and Microsoft's .NET. *In*: M. Wachowicz and L. Bodum, eds. *10th AGILE International Conference on Geographic Information Science*, 8-11th May 2007. Aalborg, Denmark: Aalborg University, p.p. 7.

Batcheller, J.K., Reitsma, F. and Gittings, B., 2007. Implementing ISO-compliant Feature Metadata. *In*: A.C. Winstanley, ed. *Proceedings of the Geographical Information Science Research UK Conference (GISRUK) 15th Annual Conference*, 11-13th April 2007. Maynooth, Republic of Ireland: National University of Ireland Maynooth, 72-78.

Batcheller, J. K., Gittings, B. M. and Dowers, S., 2007. The Performance of Vector Oriented Data Storage Structures in ESRI's ArcGIS. *Transactions in GIS,* 11(1): 47-65.

Batcheller, J.K. and Gittings, B., 2006. A GI Projects Registry for Scotland. *GeoConnexion UK,* 4 (4), 32-34.

Batcheller, J.K. and Gittings, B.M., 2006. Avenues for developing the UK's National Geospatial Metadata Service. *In*: G. Priestnall and P. Alpin, eds. *Proceedings of the Geographical Information Science Research UK (GISRUK) 14th Annual Conference*, 5 - 7th April 2006. Nottingham, UK: University of Nottingham, 259–262.

Battle, R. and Benson, E., 2008. Bridging the semantic Web and Web 2.0 with Representational State Transfer (REST). *Web Semantics: Science, Services and Agents on the World Wide Web,* 6 (1), 61-69.

Beaujardière J. d L., Mitchell H., Raskin R. and Rao A., 2000. The NASA Digital Earth Testbed. In *ACM-GIS 2000, 8th ACM Symposium on Advances in Geographic Information Systems,* 10-11 November 2000, Washington, D.C. New York, USA: ACM, 47-53.

Bédard, Y., Jeansoulin, R. and Moulin, B., 2007. Towards spatial data quality information analysis tools for experts assessing the fitness for use of spatial data. *International Journal of Geographic Information Science,* 21 (3), 261-282.

Berners-Lee, T., Hendler, J. and Lassila, O., 2001. The Semantic Web. *Scientific American,* 284 (5), 34-43.

Bertolotto, M., Carswell, J., McLoughlin, E., O'Sullivan, D. and Wilson, D., 2006. Using sketches and knowledge bases for geo-spatial image retrieval. *Computers, Environment and Urban Systems,* 30 (1), 29-53.

Bilasco, I., Gensel, J., Villanova-Oliver, M. and Martin, H., 2007. Towards Geospatial Queries in a Semantic Digital Library for 3D Data. *Transactions in GIS,* 11 (3), 337-353.

Bishr, Y., 1998, Overcoming the semantic and other barriers to GIS interoperability. *International Journal of Geographical Information Science,* 12 (4), 299-314.

Bizer, C. and Cyganiak, R., 2006. D2R Server - Publishing Relational Databases on the Semantic Web. *In: The 5th International Sematic Web Conference (ISWC2006), Athens, GA, USA, Lecture Notes in Computer Science 4273*. Berlin, Germany: Springer-Verlag, 2.

Bizer, C. and Seaborne, A., 2004. D2RQ - Treating Non-RDF Databases as Virtual RDF Graphs. *In: Proceedings of the 3rd International Sematic Web Conference (ISWC2004)*, November 7-11 2004, Hiroshima, Japan, 2.

Boucelma O. and Colonna F-M., 2005. Mediation for Online Geoservices. *In*: C Claramunt, Y-J Kwon and A Boujou eds. *Web and Wireless Geographical Information Systems: 4th International Workshop, W2GIS 2004, Lecture Notes in Computer Science 3428*. Berlin, Germany: Springer-Verlag, 81-93.

Boxall J., 2003. Geolibraries: geographers, librarians and spatial collaboration. *The Canadian Geographer,* 47 (1), 18-27.

Breeding, M., 2002. *The Open Archives Initiative* [online]. Available from: http://www.librarytechnology.org/ltg-displaytext.pl?RC=9627 [Accessed 29 March 2009].

Burrough, P.A. and McDonnell, R.A., 1998. *Principles of Geographical Information Systems.* Oxford: Oxford University Press.

Buttenfield B.P., 2002. Transmitting vector geospatial data across the Internet. *In:* M. Egenhofer, D. Mark eds. *Second International Conference on Geographic Information Science 2002, Lecture Notes in Computer Science 2478*. Berlin, Germany: Springer-Verlag, 51-64.

Catarci, T., Di Mascio, T., Franconi, E., Santucci, G. and Tessaris, S., 2003. An Ontology Based Visual Tool for Query Formulation Support. *In: Proceedings of the OTM 2003Workshops: On The Move to Meaningful Internet Systems 2003, Lecture Notes in Computer Science 2889.* Berlin, Germany: Springer, 32-33.

Clinton, W., 1994. Coordinating geographic data acquisition and access: the national spatial data infrastructure. Executive Order 12906, Federal Register 59. Washington D.C., USA: USG, 17671-17674.

Comber, A., Fisher, P. and Wadsworth, R., 2004. Integrating land-cover data with different ontologies: identifying change from inconsistency. *International Journal of Geographic Information Sciences,* 18 (7), 691-708.

Comber, A.J., Fisher, P.F. and Wadsworth, R.A., 2007. User-focused metadata for spatial data, geographical information and data quality assessments. *In*: M.

Wachowicz and L. Bodum, eds. *10th AGILE International Conference on Geographic Information Science*, 8-11th May 2007. Aalborg, Denmark: Aalborg University, p.p. 13.

Corfield, A., Dovey, M., Mawby, R. and Tatham, C., 2002, JAFER Toolkit Project - Interfacing Z39.50 and XML. *In: Proceedings of the 2nd ACM/IEEE-CS joint conference on Digital libraries, JCDL'02,* Portland, Oregon, USA. New York, USA: ACM, 289-290.

Craglia, M., Annoni, A. and Masser, I., 1999. Geographic information policies in Europe: national and regional perspectives. In *Proceedings of the EUROGI-EC Data Policy Workshop, Report EN19522 EN* 15 November 1999 Amersfoort, Netherlands. Ispra, Italy: Joint Research Centre, p.p. 28.

Craglia, M., Kanellopoulos, I. and Smits, P., 2007. Metadata: where we are now, and where we should be going. *In*: M. Wachowicz and L. Bodum, eds. *10th AGILE International Conference on Geographic Information Science*, 8-11th May 2007. Aalborg, Denmark: Aalborg University, p.p. 7.

Craven, T., 2001. *DESCRIPTION meta tags in public home and linked pages*. *LIBRES: Library and Information Science Research* 11 (2) [online]. Available from: http://libres.curtin.edu.au/LIBRE11N2/craven.htm [Accessed 29 March 2009].

Crompvoets, J., Bregt, A., Rajabifard, A. and Williamson, I., 2004. Assessing the worldwide developments of national spatial data clearinghouses. *International Journal of Geographical Information Science,* 18 (7), 665-689.

Cruz, I.F., Rajendran, A., Sunna, W. and Wiegand, N., 2002. Handling Semantic Heterogeneities Using Declarative Agreements. *In: Proceedings of the Tenth ACM International Symposium on Advances in Geographic Information Systems*, November 8-9, 2002, McLean, VA. New York, USA: ACM,168-174.

Date, C. J., 2003. *An Introduction to Database Systems, Eighth Edition*. Boston, USA: Addison Wesley.

Daukantas, P., 2003. *One site, two clicks* [online]. Available from: http://gcn.com/articles/2003/08/07/one-site-two-clicks.aspx [Accessed 29 March 2009].

Davey, A. and Murray, K., 1996. Update on the National Geospatial Database - Collaboration between Organisations. *In Proceedings of the 8th Conference of the Association for Geographic Information at GIS96*, 24-26 September 1996. Birmingham,UK. London, UK: AGI, 1-6.

Dean, D.J., 2007. Characterizing Spatial Databases via their Derivation: A Complement to Content Ontologies. *Transactions in GIS,* 11 (3), 399-412.

Delboni, T.M., Borges, K.A.V., Laender, A.H.F. and Davis Jr, C.A., 2007. Semantic Expansion of Geographic Web Queries Based on Natural Language Positioning Expressions. *Transactions in GIS,* 11 (3), 377-397.

Deng, Y., 2002. *The Metadata Architecture for Data Management in Web-based Choropleth Maps* [online]. Available from: http://www.cs.umd.edu/projects/hcil/census/JavaProto/metadata.pdf [Accessed 29 March 2009].

Devillers R., Gervais M., Bédard Y. and Jeansouin R., 2002. Spatial Data Quality: from Metadata to Quality Indicators and Contextual End-User Manual. In *OEEPE/ISPRS Joint Workshop on Spatial Data Quality Management*, 21–22 March, Istanbul, Turkey, 45-55.

Devillers, R., Bédard, Y. and Jeansoulin, R., 2005. Multidimensional Management of Geospatial Data Quality Information for its Dynamic Use Within GIS. *Photogrammetric Engineering & Remote Sensing*, 71 (2), 205-215.

Devogele T., Parent C., and Spaccapietra S., 1998. On spatial database integration. *International Journal of Geographical Information Science,* 12 (4), 335-352.

Dunfey, R.I., Gittings, B.M. and Batcheller, J.K., 2006. Towards an Open Architecture Vector GIS. *Computers and GeoSciences,* 32 (10), 1720-1732.

Easton, C., 2005. One Scotland–One Geography: A Small Country with Big Ideas. *In*: K. Fullerton ed. *The 11th EC GI & GIS Workshop ESDI: Setting the Framework*, Sardinia, Italy 29 June – 1 July 2005.

Egenhofer, M.J., 2002. Toward the Semantic Geospatial Web. *In: Proceedings of the Tenth ACM International Symposium on Advances in Geographic Information Systems*, November 8-9, 2002, McLean, VA. New York, USA: ACM, 1-4.

ESRI, 2004. *Geospatial Portal Technology* [online]. Available from: www.esri.com/library/whitepapers/pdfs/geospatial-portal-technology.pdf [Accessed 29 March 2009].

ESRI, 2002. *Metadata and GIS* [online]. Available from: www.esri.com/library/whitepapers/pdfs/metadata-and-gis.pdf [Accessed 29 March 2009].

FGDC, 1998. FGDC-STD-001-1998. Content standard for digital geospatial metadata. Reston, VA, USA: Federal Geographic Data Committee, p.p. 90.

Fileto, R., 2001. Issues on Interoperability and Integration of Heterogeneous Geographical Data. *In: Proceedings of the 3rd Brazilian Symposium on Geoinformatics (GEOINFO2001)*, 4-5 October 2001, Rio de Janeiro, Brazil, 8.

Flewelling, D.M. and Egenhofer, M.J., 1999. Using digital spatial archives effectively. *International Journal of Geographical Information Science*, 13 (1) 1-8.

Fonseca, F.T. and Egenhofer, M.J., 1999. Ontology-driven Geographic Information Systems. *In: Proceedings of the 7th ACM international symposium on Advances in geographic information systems, ACM-GIS 1999*, 2-6 November 1999, Kansas City, MO. New York, USA: ACM,14-19.

Fonseca, F. T., Egenhofer, M. J., Agouris, P. and Câmara, G., 2002. Using ontologies for integrated geographic information systems. *Transactions in GIS,* 6 (3), 231-257.

Fonseca, F. T. and Rodriguez, A., 2007. From Geo-Pragmatics to Derivation Ontologies: New Directions for the GeoSpatial Semantic Web. *Transactions in GIS,* 11 (3), 313-316.

Foy, F. B., 2001. *Metadata made easier? Development of improved online tool for "askGIraffe"*. Thesis (MSc). University of Edinburgh.

Frank A.U. 1998. Metamodels for Data Quality Description. *In:* MF Goodchild, R Jeansoulin eds. *Data Quality in Geographic Information – From Error to Uncertainty*. Paris, France:Hermes, 15-29.

Fulker, D., 2003. Metadata Strategies to Address NSDL Objectives. *In Proceedings of the 5th Russian Conference on Digital Libraries RCDL2003*, St-Petersburg, 29-31 October. St. Petersburg, Russia: St. Petersburg University, p.p. 4.

Gahegan, M., Agrawal, R., Banchuen, T. and DiBiase, D., 2007. Building rich, semantic descriptions of learning activities to facilitate reuse in digital libraries. *International Journal on Digital Libraries* 7, 81-97.

Gardels, K., 1996. *The Open GIS Approach to Distributed Geodata and Geoprocessing* [online]. Available from: http://www.ncgia.ucsb.edu/conf/SANTA_FE_CD-ROM/sf_papers/gardels_kenn/ogismodl.html [Accessed 29 March 2009].

GIgateway, 2003. Discovery Metadata Specifications. London, UK: AGI, p.p. 70.

GIgateway, 2003. GIgateway Discovery Metadata Transfer Format and Communications Protocol Specifications. London, UK: AGI, p.p. 37.

Göbel, S. and Lutze, K., 1998. Development of meta databases for geospatial data in the WWW. *In 6th International Symposium on Advances in Geographic Information Systems, ACM-GIS '98*, 6-7 November 1998 Washington D.C., USA. New York, USA: ACM, 94-99.

Goodchild, M.F. and Haining, R.P., 2004. GIS and spatial data analysis: Converging perspectives. *Papers in Regional Science,* 83 (1), 363-385.

Goodchild, M.F., Yuan, M. and Cova, T.J., 2007. Towards a general theory of geographic representation in GIS. *International Journal of Geographical Information Science,* 21 (3), 239-260.

Greenberg, J., 2003. Metadata Generation: Process, People and Tools. *Bulletin of the American Society for Information Science and Technology,* 29 (2), 16-19.

Greenberg, J., 2004. Metadata Extraction and Harvesting: A Comparrison of Two Automatic Metadata Generation Applications. *Journal of Internet Cataloging,* 6 (4), 59-82.

Greenberg, J., Spurgin, K. and Crystal, A., 2006. Functionalities for automatic metadata generation applications: a survey of metadata experts' opinions. *International Journal of Metadata, Semantics and Ontologies,* 1 (1), 3-20.

Guarino, N., 1998. Formal Ontology and Information Systems. *In: Proceedings of 1st International Conference on Formal Ontology in Information Systems (FOIS'98)*, Trento, Italy 6-8 June 1998. Amsterdam, The Netherlands: IOS Press, 3-15.

Guptill, S. G., 1999. Metadata and data catalogues. *In:* P. Longley, M. F. Goodchild, D. J. Maguire, and D. W. Rhind, eds. *Geographical Information Systems*. Chichester, UK: Wiley, 677-692.

Haining, R.P., 2003. *Spatial Data Analysis: Theory and Practice*. Cambridge: Cambridge University Press.

Halfawy, M., Vanier, D. and Froese, T., 2006. Standard data models for interoperability of municipal infrastructure asset management systems. *Canadian Journal of Civil Engineering* 33,1459-1469.

Harder, C., 1998. *Serving Maps on the Internet -Geographic Information on the World Wide Web*. Redlands, CA: ESRI.

Hart, D. and Phillips, H., 2001. *Metadata Primer - A "How To" Guide on Metadata Implementation* [online]. Available from: http://www.lic.wisc.edu/metadata/metaprim.htm [Accessed 29 March 2009].

Harvey, F. and Tulloch, D., 2006. Local-government data sharing: Evaluating the foundations of spatial data infrastructures. *International Journal of Geographic Information Science* 20 (7), 743-768.

Healey R.G. and Delve J., 2007. Integrating GIS and data warehousing in a Web environment: A case study of the US 1880 Census. *International Journal of Geographical Information Science* 21 (6), 603-624.

Herman, I., 2007. Semantic Web Adoption. *In: Proceedings of the First China Semantic Web Symposium (CSWS2007)*, 19 November 2007, Beijing, China p.p.45.

Heywood, I., 1997. *Beyond Chorley - current geographic information issues*. London, UK: AGI.

Higgins C., Medyckyj-Scott D. and Reid J. 2003. A Community Specific SDI - the Case of UK Academia. In *Geodaten- und Geodienste-Infrastrukturen - von der Forschung zur praktischen Anwendung*, June 26th-27th, Münster. Münster, Germany: University of Münster, 77-89.

Higgins, C., Robertson, A. and McGarva, G., 2005. Edinburgh University Data Library Geographic Information Standards Final Report. Edinburgh, UK: EDINA p.p. 37.

Hill L.L., Janée G., Dolin R., Frew J. and Larsgaard M., 1999. Collection Metadata Solutions for Digital Library Applications. *Journal of the American Society for Information Science,* 50 (13), 1169-1181.

Himmelstein, M., 2005. Local search: The internet is the yellow pages. *IEEE Computer* 38 (2), 26-35.

Hobona, G., James, P. and Fairbairn, D., 2004. Facilitating Data Discovery In Environmental Data Clearinghouses Through Spatial Data Mining *In*: A Lovett, ed. *Proceedings of the Geographical Information Science Research UK Conference*

(GISRUK) 12th Annual Conference, 28-30th April 2004. Norwich, UK: University of East Anglia, 163-167.

Hunter G.J. 2001. Spatial data quality revisited. *In Proceedings of GeoInfo 2001*, Rio de Janeiro, Brazil, 1–7.

Hunter, J. and Lagoze, C., 2001. *Combining RDF and XML Schemas to Enhance Interoperability Between Metadata Application Profiles* [online]. Available from: http://www.cs.cornell.edu/lagoze/papers/HunterLagozeWWW10.pdf [Accessed 29 March 2009].

ISO, 2006. BS EN ISO 19106:2006. Geographic Information—Profiles. Failand, Bristol, UK: BSi British Standards.

ISO, 2005. BS EN ISO 19109:2005 Geographic Information - Rules for Application Schema. Failand, Bristol, UK: BSi British Standards.

ISO, 2005. BS EN ISO 19115:2005. Geographic Information - Metadata. Failand, Bristol, UK: BSi British Standards.

ISO, 2007. ISO 19139:2007 Geographic Information - Metadata - XML schema. Failand, Bristol, UK: BSi British Standards.

Jenkins, C., Jackson, M., Burden, P. and Wallis, J., 1999. Automatic RDF metadata generation for resource discovery. *Computer Networks* 31, 1305-1320.

Kacmar, C., Jue, D., Stage, D. and Koontz, C., 1995. *Automatic Creation and Maintenance of an Organizational Spatial Metadata and Document Digital Library* [online]. Available from: http://www.csdl.tamu.edu/DL95/papers/kacmar/kacmar.html [Accessed 29 March 2009].

Kiehle C., 2006. Business logic for geoprocessing of distributed geodata. *Computers and Geosciences,* 32: 1746-1757.

Kim, T.J., 1999. Metadata for geo-spatial data sharing: A comparative analysis. *The Annals of Regional Science,* 33, 171-181.

Klien, E., Lutz, M. and Kuhn, W., 2006. Ontology-based discovery of geographic information services—An application in disaster management. *Computers, Environment and Urban Systems,* 30 (1), 102-123.

Kokla M. and Kavouras M. 2001. Fusion of top-level and geographical domain ontologies based on context formation and complementarity. *International Journal of Geographical Information Science,* 15 (7), 679-687.

Kuhn, W., 2001. Ontologies in support of activities in geographical space. *International Journal of Geographic Information Science,* 15 (7), 613-633.

Kuhn, W., 2005. Geospatial Semantics: Why, of What, and How? *In:* S. Spaccapietra, E. Zimányi eds. *Journal on Data Semantics III, Lecture Notes in Computer Science 3534*. Berlin, Germany: Springer, 1-24.

Lagoze, C., Sompel, H.V.d., Nelson, M. and Warner, S., 2004. *The Open Archives Initiative Protocol for Metadata Harvesting* [online]. Available from: http://www.openarchives.org/OAI/openarchivesprotocol.html [Accessed 29 March 2009].

Lathem, J., Gomadam, K. and Sheth, A.P., 2007. SA-REST and (S)mashups : Adding Semantics to RESTful Services. *In: International Conference on Semantic Computing (ICSC 2007)*, 17-19 September 2007, Irvine, California. Washington D.C., USA: IEEE, 469-476.

Laurini, R., 1994. Sharing geographic information in distributed databases. *In: Proceedings of Conference on Urban and Regional Information System Association*, Washington, DC: Urban and Regional Information Systems Association, 441-454.

Leiden, K., Laughery, K. R., Keller, J., French, J., Warwick, W. and Wood, S. D., 2001. A Review of Human Performance Models for the Prediction of Human Error.

Moffett Field, California, USA: National Aeronautics and Space Administration, p.p. 125.

Lemmens, R. and de By, R.A., 2002. Distributed GIS and metadata - Methods for the description of interoperable GIS components. *In International Workshop on Mobile and Internet GIS,* August 15-16 2002 Wuhan, China, p.p. 9.

Liddy, E., Allen, E., Harwell, S., Corieri, S., Yilmazel, O., Ozgencil, N. E., Diekema, A., McCracken, N. J., Silverstein, J. and Sutton, S. A., 2002. Automatic metadata generation and evaluation. *In Proceedings of the 25th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, 18-21 November, Tampere, Finland. New York, USA: ACM, 401-402.

Liddy, E. D., Sutton, S. A., Paik, W., Allen, E., Harwell, S., Monsour, M., Turner, A. and Liddy, J., 2001. Breaking the metadata generation bottleneck: preliminary findings. *In:* E. Fox and C. Borgman eds. *Proceedings of the 1st ACM/IEEE-CS joint conference on Digital libraries*, Roanoke, Virginia. New York, USA: ACM Press, 464.

Lim, E.-P., Liu, Z., Yin, M., Goh, D. H.-L., Theng, Y.-L. and Ng, W. K., 2005. On Organising and Accessing Geospatial and Georeferenced Web Resources using the G-Portal System. *Information Processing and Management: an International Journal* 41 (5), 1277-1297.

Limbach, T., Krawczyk, A. and Surowiec, G., 2004. Metadata Lifecycle Management with GIS Context. In *10th EC GI & GIS Workshop, ESDI State of the Art,* 23-25th June 2004 Warsaw, Poland, p.p. 10.

Longhorn, R. A., 2005. Geospatial Standards, Interoperability, Metadata Semantics and Spatial Data Infrastructure. *In Proceedings of the NIEeS Workshop on Activating Metadata* 6-7 July, Fitzwilliam College Cambridge, UK, p.p. 23.

Lowe, J.W., 2002. Finders and Keepers In Search of Spatial Data. *Geospatial Solutions*, 12 (11), 46-49.

Luo, Y., Wang, X. and Xu, Z., 2003. Extension of Spatial Metadata and Agent-based Spatial Data Navigation Mechanism. *In: Proceedings of the 11th ACM international symposium on Advances in geographic information systems GIS'03*, New Orleans, Louisiana, USA. New York, USA: ACM, 102-109.

Lutz, M. and Klien, E., 2006. Ontology-based retrieval of geographic information. *International Journal of Geographical Information Science* 20 (3), 233-260.

Lutz, M. and Kolas, D., 2007. Rule-Based Discovery in Spatial Data Infrastructure. *Transactions in GIS* 11 (3), 317-336.

Lynch, C.A., 2001. *Metadata Harvesting and the Open Archive Initiative - ARL Bimonthly Report 217* [online]. Available from:

http://www.arl.org/resources/pubs/br/br217/br217mhp.shtml [Accessed 29 March 2009].

Maguire, D.J., 1991. An overview and definition of GIS. In: D.J. Maguire, M.F. Goodchild and D.W. Rhind eds. *Geographical Information Systems: Principles and Applications.* London, UK: Longman, 9-20.

Maguire D.J. and Longley P.A., 2005. The emergence of geoportals and their role in spatial data infrastructures. *Computers, Environment and Urban Systems,* 29, 3-14.

Mäkelä, E., Hyvönen, E. and Saarela, S., 2006. Ontogator — A Semantic View-Based Search Engine Service for Web Applications. *In: Proceedings of the 5th International Sematic Web Conference (ISWC 2006), Athens, GA, USA Lecture Notes in Computer Science 4273.* Berlin, Germany: Springer, 847-860.

Masser, I., 1999. All shapes and sizes: the first generation of national spatial data infrastructures. *International Journal of Geographical Information Science,* 13 (1), 67-84.

Masser, I., 2005. *GIS Worlds - Creating Spatial Data Infrastructures*. Redlands, USA: ESRI.

Mathys, T., 2004. The Go-Geo! Portal Metadata Initiatives. *In*: A Lovett, ed. *Proceedings of the Geographical Information Science Research UK Conference*

(GISRUK) 12th Annual Conference, 28-30th April 2004. Norwich, UK: University of East Anglia, 148-154.

Medyckyj-Scott, D., Chappell, C., Pradhan, A. and O'Hanlon, C., 2001. A geo-spatial data resource discovery tool for UK Further and Higher Education - Project Overview and Recommendations. Edinburgh, UK: EDINA, p.p. 60.

Moellering, H., 1992. Opportunities for Use of the Spatial Data Transfer Standard at the Sate and Local Levels. *Cartography and Geographic Information Systems,* 19 (5), 332-334.

Movva, S., Ramachandran, R., Li, X., Khaire, S., Keiser, K., Conover, H. and Graves, S., 2005. Syntactic and semantic metadata integration for science data use. *Computers and Geosciences,* 31 (9), 1126-1134.

Nanson, B., Smith, N. and Davey, A., 1995. What is the British National Geospatial Database? *In 7th Conference of the Association for Geographic Information*, 22 November 1995, Birmingham, UK. London, UK: AGI, 1-8.

Nebert, D.D., 2004. *Developing Spatial Data Infrastructures: The Spatial Data Infrastructure Cookbook v2.0* [online]. Available from: http://www.gsdi.org/gsdicookbookindex.asp [Accessed 29 March 2009].

Nebert, D.D., 2004. *Technical Baseline for Accessing a Virtual Global SDI* [online]. Available from: http://gsdidocs.org/gsdiconf/GSDI-7/papers/FTdn.pdf [Accessed 29 March 2009].

NGDF, 1999. Communication Protocols for a Distributed Geospatial Metadata Service. London, UK: AGI, p.p. 15.

Nikolaos, A., Kostas, K., Michail, V. and Nikolaos, S., 2005. The Emerge of Semantic Geoportals. *In: Proceedings of the OTM Workshops: On The Move to Meaningful Internet Systems 2005, Lecture Notes in Computer Science Vol. 3762*. Berlin, Germany: Springer, 1127-1136.

Nogueras-Iso, J., Zarazaga-Soria, F.J., Béjar, R. and Álvarez, P.J. 2005. OGC Catalog Services: a key element for the development of Spatial Data Infrastructures. *Computers and Geosciences,* 31, 199-209.

Nogueras-Iso, J., Zarazaga-Soria, F.J., Lacasta, J. and Muro-Medrano, B.R., 2004. Metadata standard interoperability: application in the geographic information domain. *Computers, Environment and Urban Systems,* 28, 611-634.

OGC, 2004. Geospatial portal reference architecture: a community guide to implementing standards-based geospatial portals. Wayland, MA: OGC, p.p. 17.

OGC, 2007. OGC Catalogue Services Specification 2.0.2. Wayland, MA: OGC, p.p. 204.

OGC, 2003, OpenGIS Reference Model (OGC 03-040). G. Percivall (Ed.), p. 108Open Geospatial Consortium (OGC)).

Passin, T. B., 2004. Explorer's Guide to the Semantic Web. Greenwich, CT, USA: Manning.

Peng Z-R., 2005. A proposed framework for feature-level geospatial data sharing: a case study for transportation network data *International Journal of Geographical Information Science,* 19 (4), 459-481.

Peng, Z.-R. and Tsou, M.-H., 2003. *Internet GIS - Distributed Geographic Information Services for the Internet and Wireless Networks*. Hoboken, NJ, USA: Wiley and Sons.

Pfoser, D., Pitoura, E. and Tryona, N., 2002. Metadata Modeling in a Global Computing Environment. *In: Proceedings of the 10th ACM International Symposium on Advances in Geographic Information Systems (ACM-GIS2002)* McLean, VA. New York, USA: ACM 68-73.

Pierre, M.S. and LaPlant-Jr, W.P., 1998. *Issues in Crosswalking Content Metadata Standards* [online]. Available from:

http://www.niso.org/publications/white_papers/crosswalk/ [Accessed 29 March 2009].

Piwowar, J.M., Ledrew, E.F. and Dudycha, D.J., 1990. Integration of spatial data in vector and raster formats in a geographic information system environment. *International Journal of Geographic Information Science,* 4 (4), 429-444.

Plewe, B., 1997. *GIS online: Information retrieval, mapping, and the Internet*. Santa Fe: Onward Press.

Pradhan, A., 1999. *The Geospatial Metadata Server: Online Metadata Database, Data Conversion and GIS Processing*. Thesis (PhD). University of Edinburgh.

Probst, F., Ginotti, F., Morantes, A., Esbri, M., Barros, M., Gutierrez, M. and Kuhn, W., 2004. Connecting ISO and OGC Models to the Semantic Web. In: *Proceedings of the Third International Conference on Geographic Information Science,* Adelphi, MD, USA, p.p. 5.

Putz, S., 1994. Interactive information services using world-wide web hypertext. *Computer networks and ISND Systems*, 27 (2), 273-280.

Qi, L., Lingling, G., Feng, H. and Yong, T., 2004. A Unified Metadata Information Management Framework for Digital City. *In: Proceedings of IEEE's Geoscience and*

Remote Sensing Symposium, 20-24 September 2004 Anchorage, Alaska, USA. Washington D.C., USA: IEEE, 4422-4424.

Rackham, L., 2004. *An Independent Review of the Sustainability of a UK Metadata Service for Geographically Related Information*. London, UK: AGI / OS, p.p. 49.

Rajabifard, A., Binns, A., Masser, I. and Williamson, I., 2006. The role of sub-national government and the private sector in future spatial data infrastructures. *International Journal of Geographic Information Science,* 20 (7), 727-741.

Rajabifard, A., Feeney, M. E., Williamson, I. and Masser, I., 2003. National SDI-initiatives. *In:* I. P. Williamson, A. Rajabifard, M. F. Feeney eds. *Developing Spatial Data Infrastructures: From Concept to Reality.* London, UK: Taylor & Francis, 95-109.

Reitsma, F. and Hiramatsu, K., 2006. Exploring GeoMarkup on the Semantic Web. In: J. Suarez and B Markus eds. *Proceedings of the 9th International AGILE Conference on Geographic Information Science,*Visegrád, 20-22 April, 2006. Visegrád, Hungary: Unversity of West Hungary, 110-118.

Rhind, D., 1977, Computer-aided cartography. *Transactions Institute of British Geographers*, 2 (1), 71-96.

Rhind, D., 1997. Overview of the National Geospatial Data Framework. *In Proceedings of the Association for Geographic Information – GIS Conference 1997 (AGI - GIS'97)*. London, UK: NGDF, 1-14.

Richardson, L. and Ruby, S., 2007. RESTful Web Services. Sebastopol, USA: O'Reilly Media, Inc.

Rocha, J. G. and Henriques, P. R., 2004. Towards XML Web Services based Clearinghouses. *In 7th Global Spatial Data Infrastructure Conference*, 2-6 February 2004. Bangalore, India, p.p. 14.

Sayar, A., Pierce, M.E. and Fox, G.C., 2005. *Developing GIS Visualization Web Services for Geophysical Applications* [online]. Available from: http://grids.ucs.indiana.edu/ptliupages/publications/isprs_asayar.pdf [Accessed 29 March 2009].

Schuurman, N. and Leszczynski, A., 2006. Ontology-based metadata. *Transactions in GIS* 10 (5), 709-726.

Simeoni, F., (2004). The Case for Metadata Harvesting. *Library Review*, 53 (5), 255-258.

Smith, B. and Mark, D.M., 2001. Geographical categories: an ontological investigation. *International Journal of Geographic Information Science,* 15 (7), 591-612.

Soete, G.J., 1997. *Transforming Libraries - Issues and Innovations in Geographic Information Systems, Volume 2*. Washington DC, USA: Association of Research Libraries.

Stephens, R. and Hochgurtel, B., 2002. Visual Basic .NET and XML. New York, USA: John Wiley and Sons Inc.

Swan, A. and Awre, C., 2006. *Linking UK Repositories: Technical and organisational models to support user-oriented services across institutional and other digital repositories* [online]. Available from: http://www.jisc.ac.uk/media/documents/programmes/digitalrepositories/linkingukrepositoriesreport.pdf [Accessed 29 March 2009].

Tait, M., 2005. Implementing geoportals: applications of distributed GIS. *Computers, Environment and Urban Systems,* 29, 33-37.

Tang, W., 2004. *Building Spatial Portals: Key Issues and Lessons Learnt* [online]. Available from: http://www.gisdevelopment.net/application/miscellaneous/ma04076pf.htm [Accessed 29 March 2009].

Taylor, M., 2004. Metadata - Describing geospatial data. *In:* D.D. Nebert, D. D. ed. *Developing Spatial Data Infrastructures: The SDI Cookbook version 2.0*, 24-38. [online]. Available from: http://www.gsdi.org/gsdicookbookindex.asp [Accessed 29 March 2009].

Troll, D. and Moen, B., 2001. *Report to the DLF on the Z39.50 Implementers' Group - Moving Towards the Future of Z39.50. Issues and Options Based on ZIG Meeting Discussions December 6-7, 2000* [online]. Available from: http://www.diglib.org/architectures/zig0012.htm [Accessed 29 March 2009]

Tsou, M.-H., 2002. An Operational Metadata Framework for Searching, Indexing, and Retrieving Distributed Geographic Information Services on the Internet. *In:* M. Egenhofer and D. Mark eds. *Geographic Information Science (GIScience 2002)*. Berlin, Germany: Springer-Verlag, 313-332.

Tulloch D.L. and Robinson M., 2000. A progress report on a U.S. National Survey of Geospatial Framework Data. *Journal of Government Information,* 27, 285-298.

Turner A., 2006. *Introduction to Neogeography* [online]. O'Reilly Press. Available from: http://www.oreilly.com/catalog/neogeography [Accessed 29 March 2009]

Turner, F., 1997. *An Overview of the Z39.50 Information Retrieval Standard* [online]. Available from: http://www.ifla.org/VI/5/op/udtop3/udtop3.htm [Accessed 29 March 2009].

Tyler, G. J., 2002. *Managing Metadata: Developing technical solutions for the askGIraffe geospatial metadata gateway*. Thesis (MSc). University of Edinburgh, UK.

Vermeij, B., 2001. *Implementing European Metadata Using ArcCatalog - ArcUser* [online]. Available from: http://www.esri.com/news/arcuser/0701/metadata.html [Accessed 29 March 2009].

Visser, U., Stuckenschmidt, H., Schuster, G. and Vögele, T., 2002. Ontologies for geographic information processing. *Computers and Geosciences,* 28 (1), 103-117.

West-Jr., L. A. and Hess, T. J., 2002. Metadata as a knowledge management tool: supporting intelligent agent and end user access to spatial data. *Decision Support Systems,* 32, 247-264.

Westbrooks E.L., 2004. Distributing and Synchronizing Heterogenous Metadata for the Management of Geospatial Information Repositories for Access. *In:* D Hillmann and E Westbrooks eds. *Metadata in Practice: Building the Diverse Digital Library*. Chicago, USA: American Library Association, 139-157.

Wiegand, N. and García, C., 2007. A Task-Based Ontology Approach to Automate Geospatial Data Retrieval. *Transactions in GIS,* 11 (3), 355-376.

Xie, R. and Shibasaki, R., 2007. Imagery Metadata Development Based on ISO/TC 211 Standards. *Data Science Journal* 6, 28-45.

Yang, B., Purves, R., and Weibel, R., 2007. Efficient transmission of vector data over the Internet. *International Journal of Geographical Information Science,* 21 (2), 215-237.

Yang, C. P., Wong, D. W., Yang, R., Kafatos, M. and Li, Q., 2005. Performance-improving techniques in web-based GIS. *International Journal of Geographical Information Science,* 19 (3), 319-342.

Yuan, M., Buttenfield, B., Gahegan, M.N. and Miller, H., 2004. Geospatial Data Mining and Knowledge Discovery. In: R. McMaster and L. Usery eds. *Research Challenges in Geographic Information Science*. Boca Raton, FL, USA: CRC Press.

Zhao, P., Chen, A., Liu, Y., Di, L., Yang, W. and Li, P., 2004, Grid Metadata Catalog Service-Based OGC Web Registry Service [online]. Available from: http://laits.gmu.edu/Papers/Peisheng_Aijun.pdf [Accessed 29 March 2009].

Appendix B – Peer reviewed papers

Avenues for developing the UK's National Geospatial Metadata Service – GISRUK 2006

A GI Projects Registry for Scotland – GEOconnexion UK September/October 2006

Implementing ISO-compliant Feature Metadata – GISRUK 2007

Automating geospatial metadata using ESRI's ArcGIS and Microsoft's .NET – AGILE 2007

Automating geospatial metadata generation—An integrated data management and documentation approach – Computers and GeoSciences 34 2008

Avenues for developing the UK's National Geospatial Metadata Service

James K Batcheller, Bruce M Gittings

Institute of Geography, School of GeoSciences University of Edinburgh,
Drummond Street, Edinburgh EH8 9XP, Scotland, U.K
Tel. + 44 131 650 2565 Fax + 44 131 650 2524
{jk.batch; bruce}@ed.ac.uk

KEYWORDS: geospatial metadata service, gigateway, metadata

1. Introduction

The state of public sector geospatial data sharing and exchange in the UK, as facilitated by the *gigateway* service, is currently at a crossroads. Ambiguities surrounding its purpose, technological expectations, ongoing source of funding (as currently enshrined within the NIMSA[1] agreement, Rackham, 2004) coupled with doubts as to whether the AGI[2] shall continue to act as custodian have led to the national geospatial metadata service facing a somewhat uncertain future.

And yet further challenges exist. With the rapid and ongoing evolution of geographical information software and services offered over the Internet and World Wide Web - from sophisticated web mapping and routing applications (Elgin, 2005) to regional, national and global Spatial Data Infrastructure (R/N/GSDI) initiatives and commercially maintained GIS Portals - it can be reasoned that end-user expectations have also evolved, arguably passed what the gigateway service can offer currently.

It is in this light that the timeliness of a re-examination of how public geospatial metadata is published in the UK via the gigateway service is argued. The current period of reflection provides a suitable setting in which to investigate means of building upon the goodwill accrued by the service to date, and where its future strategy can be deliberated. A number of pragmatic short-term solutions are presented in the current research; it is also intended to fuel the debate as to how to move the gigateway service forward in the longer term.

2. The gigateway service

The gigateway service functions through the support of a distributed web-based network focussed on serving geospatial metadata – metadata designed to provide a means of identifying where the geospatial assets it describes may be located. Users submit metadata queries via a central gateway, essentially a web-based form that accepts keywords and geographic extents as search parameters. Processed queries are sent to remote clients (nodes) that expose the geospatial metadata of participating organisations and return search results to the gateway, where they are displayed. Metadata items identified as surrogates for datasets sought provide details of where such datasets may be found.

3. Developing gigateway

Ever since the rollout of the first functioning national metadata exchange initiative in July 2000, work towards enhancing the quality of service has been ongoing – improving node reliability and the provision of metadata authoring utilities being notable examples. Contributions of this nature have fuelled the success of the service (in all its guises), resulting in the recognition of gigateway as an important component in the UK's GIS sector today. Nevertheless, scope remains for further work that

[1] National Interest Mapping Service Agreement

[2] Association for Geographic Information

would clearly serve to consolidate the service's standing. Considering such issues, it is perhaps useful to think in terms of an ideal (potentially unrealistic) scenario, how the current service operates, and how the current service can be developed to bridge the gap in between.

3.1 Geospatial metadata generation

<u>Ideal scenario</u>
The preparation of a geospatial dataset destined for exposure to the wider GIS community is accompanied by the automatic generation of "publication ready" geospatial metadata documenting the asset. The metadata is validated according to the desired discovery metadata standard in the process.

<u>Current service</u>
Once geospatial assets are identified as candidates for exposure via surrogates within the current service, any existing metadata elements must undergo supervised transformation so as to conform to gigateway's discovery metadata specification, with the metadata record being completed with manual entries.

<u>Potential bridging strategy</u>
An option to streamline metadata production is presented though an automated metadata generation tool in the form of a GIS application plug-in. The proposed functionality is as follows: the geospatial asset's inherent metadata is automatically extracted, complemented with pre-defined author metadata stored within the system and finally manually completed (in accordance with the UK GEMINI metadata standard profile) with a limited number of descriptive metadata elements.

<u>Benefits of bridging strategy to gigateway</u>
The generation of metadata is frequently low on the list of and organisations' priorities, and even when undertaken, is often viewed as being time-consuming and tedious. Streamlining metadata creation by automating certain stages will alleviate such a perception, enable more rapid publication of geospatial assets, enhance the worth of the service with increased metadata volumes and facilitate tapping under-exploited data resources.

3.2 Geospatial metadata integrity

<u>Ideal scenario</u>
A single geospatial metadata record depicts a single geospatial asset, eliminating redundancy and safeguarding metadata integrity.

<u>Current service</u>
Multiple instances of metadata records currently exist per geospatial asset, often in multiple locations and in multiple formats. Strategies for data management range from those which maintain metadata records alongside the data they describe; others maintain them separately, whether as flat files (XML, CSV or HTML) or within a RDBMS. Updates are clearly not straightforward, with a high potential for inconsistency.

<u>Potential bridging strategy</u>
A utility to automate the update of all instances of a metadata record would address integrity concerns. The fundamental concept here is of a master copy that is edited, with updates cascaded throughout other instances of the record. The method of update is somewhat dictated by how the record is manifested elsewhere, with appropriate options including RDBMS scripting, element level editing and file synchronisation.

<u>Benefits of bridging strategy to gigateway</u>
The consistent provision of quality geospatial metadata is important in maintaining the confidence of the participating community. Enforcing metadata integrity with specifically designed tools serves to safeguard such quality.

3.3 Node installation

Ideal scenario
The software infrastructure necessary to manage and publish geospatial metadata is readily installed and easily configured.

Current service
A number of distinct and unique software solutions are installed and configured discretely on each node with little or no interconnection. Setup is frequently left to IT departments, outsourced to consultancies or indefinitely postponed.

Potential bridging strategy
Bundling the open source components of a generic node setup within an easily executed installation program would empower a non-specialist, user driven installation. The concept here is to enable the configuration of metadata serving software during the install process, combining all further operations necessary for exposing metadata into one user-friendly process stream.

Benefits of bridging strategy to gigateway
By simplifying the process necessary for exposing geospatial metadata, the barrier to contribution amongst potential community participants is lowered. Potential gains are offered in terms of metadata volumes and the tapping of geospatial resources via the empowerment of smaller organisations that would otherwise be unable or unwilling to mount their own node.

3.4 Path to geospatial data

Ideal scenario
The geospatial metadata record provides a direct link to the data it describes, enabling immediate visualisation, purchase (if copyrighted) and download.

Current service
Geospatial metadata records currently served by gigateway contain contact information for the data custodian in such forms as telephone numbers, e-mail addresses and website URLs. Once a request has been submitted, the custodian must locate the appropriate dataset(s); consumers are forced to wait for a response. As visual representations of the datasets are unavailable, there is scope for requesting inappropriate data, which causes further delay.

Potential bridging strategy
Linking metadata to a representation of the data it describes will ensure selection of the appropriate dataset. Procurement is enabled by facilitating immediate download for free datasets, with a provision for the download of copyrighted material by facilitating on demand purchases via online transaction services.

Benefits of bridging strategy to gigateway
Providing a visual preview of data ensures selection of suitable geospatial assets and enhances working efficiency. By providing further facilities for downloading data, workflows are similarly enhanced by reducing the time cost of data procurement. Offering this service under the aegis of the gigateway service allows a better overview of the effectiveness of the overall service as exchanged data volumes can be more accurately calculated.

4. Conclusion and Future Work

Thus the case for elaborating a strategy for furthering the UK's national geospatial metadata service is presented, together with proposed development options. Such options may be categorised by either the enhancement of the current model by focussing on improving on its efficiency, or by the extension of the current model by coupling further solutions to it. A third option, whilst not treated in depth here, must also be considered – that of total model replacement.

Perhaps more relevant to the elaboration of a long-term strategy, replacing the current service model with for instance an all-encompassing geospatial data architecture such as that elaborated by the OGC[3] (OGC, 2004) presents intriguing possibilities, as well as complications. Implementing an open and modular architecture would provide the security of future-proofing the service, and yet provide a vehicle with the potential for offering an integrated geospatial data infrastructure with portrayal, cataloguing and portal functionality.

Potential complications meanwhile not only include the backward compatibility of any new paradigm with node hosts' own incumbent solutions and such service extensions as discussed above, but also the need to resolve political issues such as the need for consensus amongst participating bodies, sourcing funds for technical migration and not least of all, agreement as to the future direction of the initiative.

5. Acknowledgements

The authors would like to thank the members of gigateway's Advisory Board for their cooperation and assistance during the course of the current research. They would also like to thank past members of the NGDF and AskGIraffe teams for their insights and advice.

---

[3] Open Geospatial Consortium

References

Elgin, B (2005), Google Tops the Charts. In BusinessWeek Online, November 30, 2005, http://www.businessweek.com/technology/content/nov2005/tc20051130_704115.htm

Open Geospatial Consortium Inc (2004), Geospatial Portal Reference Architecture: A Community Guide to Implementing Standards-Based Geospatial Portals OGC Draft Report No OGC 04-039, http://www.opengeospatial.org/

Rackham, L (2004), An Independent Review of the Sustainability of a UK Metadata Service for Geographically Related Information. London, AGI / OS**:** 1-49

Biography

James Batcheller is currently in his first year as a PhD student in the School of GeoSciences in the University of Edinburgh. He has a background in IT and Environmental Sciences and has recently returned to his studies after a number of years in the GIS industry.

Bruce Gittings is a member of the Edinburgh Earth Observatory (University of Edinburgh) involved in GIS teaching and research. His particular interests include database management and web-based GIS.

A GI Projects Registry for Scotland

James Batcheller and Bruce Gittings outline the evolution of a new online resource to help researchers find places, spaces and faces North of the border

Geographical Information systems, services and related data have long since proliferated beyond the confines of the specialist domains in which they were first devised. Public, private and academic organisations not traditionally versed in geospatial practices have become increasingly reliant upon GI disciplines to help deliver products, provide services and drive research. The result has been not only an explosion in the volumes of data produced, but also the associated tools, applications and accompanying expertise, spread across a range of disciplines.

Partnerships between private, local and central government agencies and research bodies have been key in driving such developments. This wider involvement nevertheless comes with its own implications, especially when considering the diversity of prospective collaborators.

Disparities in data format and information exchange protocols necessitate the use of collating techniques at destination, costing time, resources and presenting the potential for loss of detail in the information being transferred.

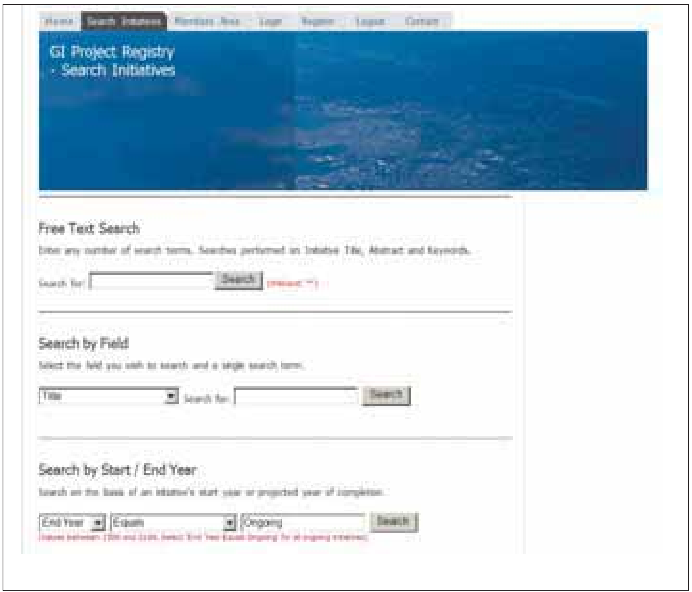
Identifying and contacting appropriate domain specialists for expert counsel, meanwhile, is time-consuming at best where no tangible avenues of communication already exist. Equally, few tasks are truly unique, but identifying appropriate partners who may have achieved portions of one's own goals can be difficult. Collaborative efforts, therefore, offer potential savings in time, effort and resources, and are most effective when underpinned by mechanisms and protocols to manage and exchange data, expertise and other related geographic information in a coherent, integrated and effective manner.

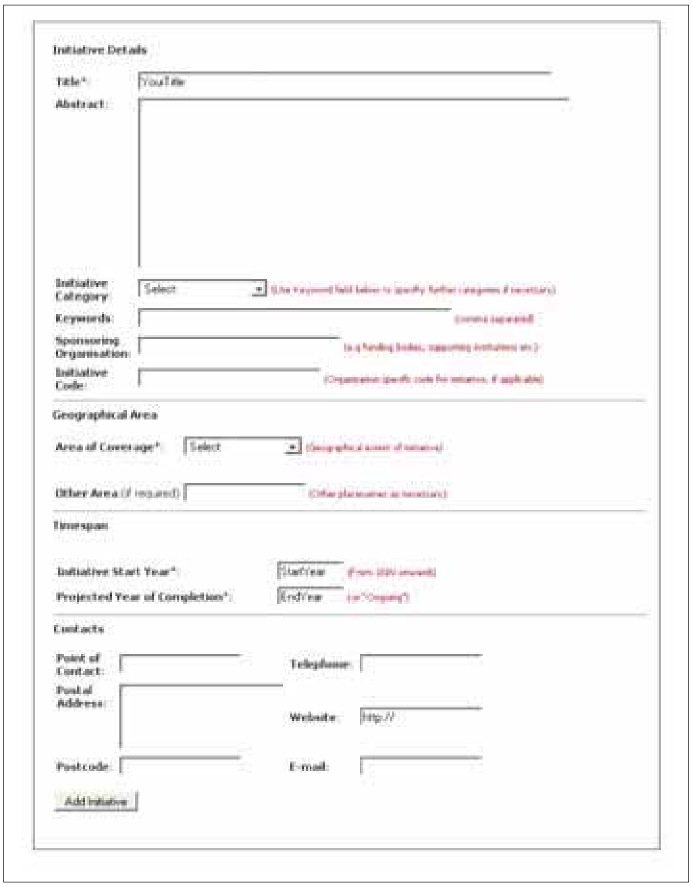
"One Scotland - One Geography"

It was in recognition of these issues that in 2005 the Scottish Executive, supported by the Association for Geographic Information (Scotland), launched the GI Strategy for Scotland titled "One Scotland - One Geography". Providing a framework in which detailed policies can be developed, the Strategy sets out to promote as wide use of GI as possible, supporting the faces, places and spaces of Scotland.

Endorsed by a broad range of stakeholders from the public, private and academic sectors, the Strategy outlines a five-year roadmap for

> "Scotland is a small country, where strong partnerships and shared approaches not only make good sense, but are achievable. We need to develop common approaches, common standards and the capacity to share as much as possible of the resulting data and information, to ensure a joined up response to peoples' needs."
>
> **Tom McCabe**, MSP Minister for Finance and Public Service Reform





Pictured clockwise from top left: The GI Project Registry Homepage; search page; form for adding an initiative; Members Area

action commencing in 2006, intending to promote efficient government through the re-use of ideas, data and systems. Drawn up by the Executive, specific goals include:

- provision of strategic vision and leadership to ensure an inclusive, co-ordinated approach to GI in Scotland
- delivery of accurate, up-to-date GI
- development of avenues through which GI can be shared
- promotion of the benefits of GI across all sectors
- promotion of the appropriate technical/professional standards for the efficient and effective use of GI in Scotland

While the initial focus is on central government and infrastructural development, the need for continued involvement of the wider GI community remains a central tenet for continued success. For such involvement to be effective, opportunities for improved engagement between community members must be identified and promoted. Wider visibility of GI-related activity across Scotland is necessary, not only to avoid duplication of effort/misdirection of resources, but to facilitate knowledge transfer, the cross-fertilisation of ideas and to provide a showcase for such activities.

The GI Project Registry

It was with this in mind that the Edinburgh Earth Observatory - the locus for GI and Remote Sensing research at the University of Edinburgh - in collaboration with the Scottish Executive and AGI-Scotland, developed an online directory providing details of GI-related projects, implementations and research activities.

The 'GI Project Registry' (www.gisprojects.net) aims to help identify and promote synergies in step with the GI Strategy for Scotland's overall objectives above. Providing a means in which initiatives may be recorded and searched, participants are not only able to publicise their own activities, but also gain an insight into what is going on elsewhere in the wider community.

By offering such a virtual 'shop window' it is intended to encourage dissemination of expertise, identify opportunities for collaboration and forge lasting linkages across all GI related sectors. As a user-driven service, it is intended that the site be self-sustaining, with responsibility falling on each participant to maintain and update their own records.

Developing the site

One of the principal specifications of the Registry was that it be quick to develop and easy to maintain - a dedicated server running open-source Linux was accordingly set up and deployed.

The need for an interactive, searchable and flexible resource was quickly recognised and a database-driven web site specified. This is also based on open-source components, conforming to the AMP model: Apache acts as the web document server to accept user requests and transmit the HTML pages; MySQL serves as the underlying relational database system to store all the data and PHP operates as the scripting language through which the components are inter-connected.

Development of the Registry proceeded in tandem with a study performed by the Scottish Executive seeking to identify stakeholders willing to participate. Once identified, each stakeholder was contacted in turn, informed as to the purpose of the Registry, and asked for information on any GI-related activity they deemed appropriate for inclusion.

The definition of a 'GI-related activity' was deliberately open and non-prescriptive to encourage as broad an involvement as possible, although it was made clear that dataset-specific information should be contributed to a geospatial metadata service such as gigateway (www.gigateway.org.uk). Data collected during the study was used to populate the website prior to launch not only to provide a service with immediate value but to encourage further contributions from users presumed more likely to participate on discovery of a fully-functional solution.

Members Area

Accounts were assigned to stakeholders following the data collection phase; subsequent users of the site willing to partake are prompted to register prior to contributing their own initiatives. Once registered, users gain access to the site's Members Area, enabling management of account specific information. Here, contributors can update their account details, add and edit their own initiatives as well as review contributions made.

It was recognised that the number of fields used to describe an initiative should be kept to a minimum to encourage contribution and to prevent it from being cumbersome. The proposed fields were agreed with the stakeholders they were sufficiently comprehensive to be meaningful. Fields marked as mandatory were similarly kept to a minimum, although users are encouraged to provide as many details as is possible.

Entries provide the basic details of each initiative, its geographical area of coverage, its projected or realised time span and points of contact. Once recorded, details are managed by the contributing user (or an assigned delegate) who effectively becomes the record's custodian. Automated methods of preventing records becoming stale will be reviewed as part of the project.

Identifying linkages

Site access is not limited to those registered - anonymous users may browse and search recorded initiatives at will. Whatever the case, it is using the power of the site's search facility that linkages between initiatives can be identified.

Free text search may be performed on the Title, Abstract or Keyword fields, or limited to specific fields within the database.

Specific fields such as Initiative Category or Sponsoring Organisation for instance may be used in queries to locate initiatives within similar domains; queries on Area of Coverage and Local Authority will uncover those occupying the same geographic space, whereas those on Initiative Start Year and Projected Year of Completion will pinpoint activities that are currently active or of historical interest. Contact details for each record are provided, offering an appropriate point of contact should further information be required and facilitating internetworking between participants without the need for an intermediary.

A First Step

The 'GI Project Registry' is seen as an initial, yet important, first step underpinning the GI Strategy for Scotland, with emphasis on the resources critical to the Strategy's success - people, their expertise and how they may be linked to provide an invaluable resource.

Progress in other areas of the Strategy continues to be made, with ongoing efforts to promote the benefits and coordinated provision of GI data and services gaining significant traction amongst the wider Scottish geospatial community. And while the Strategy - and the Project Registry - are primarily intended for this Scottish audience, both have and will continue to be developed with a view to encouraging participation from the wider UK community, and beyond.

Acknowledgements

The authors acknowledge the considerable contribution of Gillian Dalrymple, Cameron Easton, Fiona MacDonald and Ian Elliott at the Scottish Executive to the design and population of the GI Projects Registry.

James K Batcheller and **Bruce M Gittings** are with the Edinburgh Earth Observatory at the University of Edinburgh *(www.geos.ed.ac.uk/research/eeo/)* and can be contacted by email at jkbatch@gmail.com and bruce@geo.ed.ac.uk respectively

Implementing ISO-compliant Feature Metadata

James K Batcheller, Femke Reitsma, Bruce M Gittings

Edinburgh Earth Observatory, Institute of Geography, School of GeoSciences
University of Edinburgh, Drummond Street, Edinburgh EH8 9XP, Scotland, UK
Telephone: +44 131 650 2565 Fax: +44 131 650 2524
{jk.batch|femke.reitsma|bruce}@ed.ac.uk
http://www.eeo.ed.ac.uk/

1. Introduction

The role metadata have played in the management of geospatial datasets has been widely documented. Often employed by institutions to organise, maintain and document their geographic resources internally, metadata can also provide a vehicle for exposing marketable data assets externally when contributed to geospatial exchange initiatives such as the UK's public sector metadata service *gigateway* and its academic equivalent *Go-Geo!* Regardless of application, geospatial metadata provide information which support decisions involving the resources they describe, whether for the data custodian, data user or potential procurer.

The evolution of geographic data storage and access strategies have meanwhile resulted in the introduction of solutions advancing beyond the traditional monolithic, single-user dataset paradigm to multi-user enterprise or corporate GIS solutions that support access and extraction of information at sub-dataset levels. These advances have more recently been mirrored by the development of feature-driven web visualisation techniques such as OGC-compliant Web Feature Services (WFS) in a fundamental departure from the pioneering raster image approaches.

Consequent to these developments, not only have multiple (and often simultaneous) routes of access to individual data resources been opened, but the ability to disseminate and exchange resource subsets has been enabled. Nevertheless, with no guarantee that dataset metadata (when extant) will either accompany these subsets, or indeed be adequate to accurately depict their key statistics, doubts can arise as to the appropriateness of the constituent features. It is argued that many affected applications would benefit from even a minimal indication of quality, embedded at the feature level.

2. ISO Standards and Feature Metadata

An integral part of any metadata approach should be the adoption of a well-defined convention. The ISO 19115:2005 Geographic Information – Metadata standard details a content schema for the documentation of geospatial data. While some provision has been made for the generation of metadata at varying degrees of resolution, its focus arguably lies with the depiction of data at the dataset, and to a lesser degree, dataset series level. Metadata at the sub-dataset level are presented within a metadata hierarchy (Figure 1); but suggested implementations only include definitions at these levels when exceptions occur. Further, as metadata conforming to ISO 19115 are held discrete from the resources they describe by convention, this treatment for metadata on its own is insufficient when considering some of the aforementioned issues.

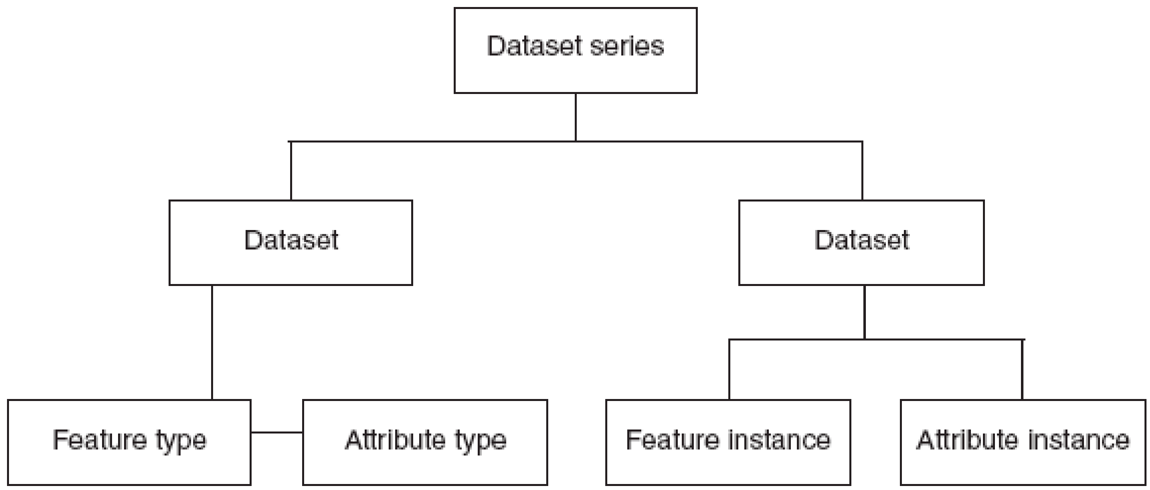


Figure 1. ISO 19115 Metadata Hierarchy

The ISO 19109:2005 Geographic Information – Rules for Application Schema standard meanwhile allows for the definition of conceptual data models which define the logical structure of an application's data. Geographic feature types are classified based on a structure defined by the General Feature Model (GFM); feature type definitions (detailing feature attributes, operations and association roles) may be elaborated in feature catalogues.

Of particular interest is its specific treatment for feature attributes as well as the general ability to integrate within any ISO 10109 application schema other ISO standard schemas. Here, any feature attribute (GF_AttributeType) can have atomic metadata items associated with it by sub-classing entities beneath the GF_QualityAttributeType specialisation of the GF_MetadataAttributeType entity (Figure 2). Attribute types accordingly defined (specifically, to carry metadata information such as quality and currentness) obtain their value type definitions and value domains from the ISO 19115 MD_Metadata entity. It should be noted however that temporal attributes used as metadata items (described below) are arrived at in this manner, and not through the GF_TemporalAttributeType entity included in Figure 2, which is employed for conventional temporal attribute definition.

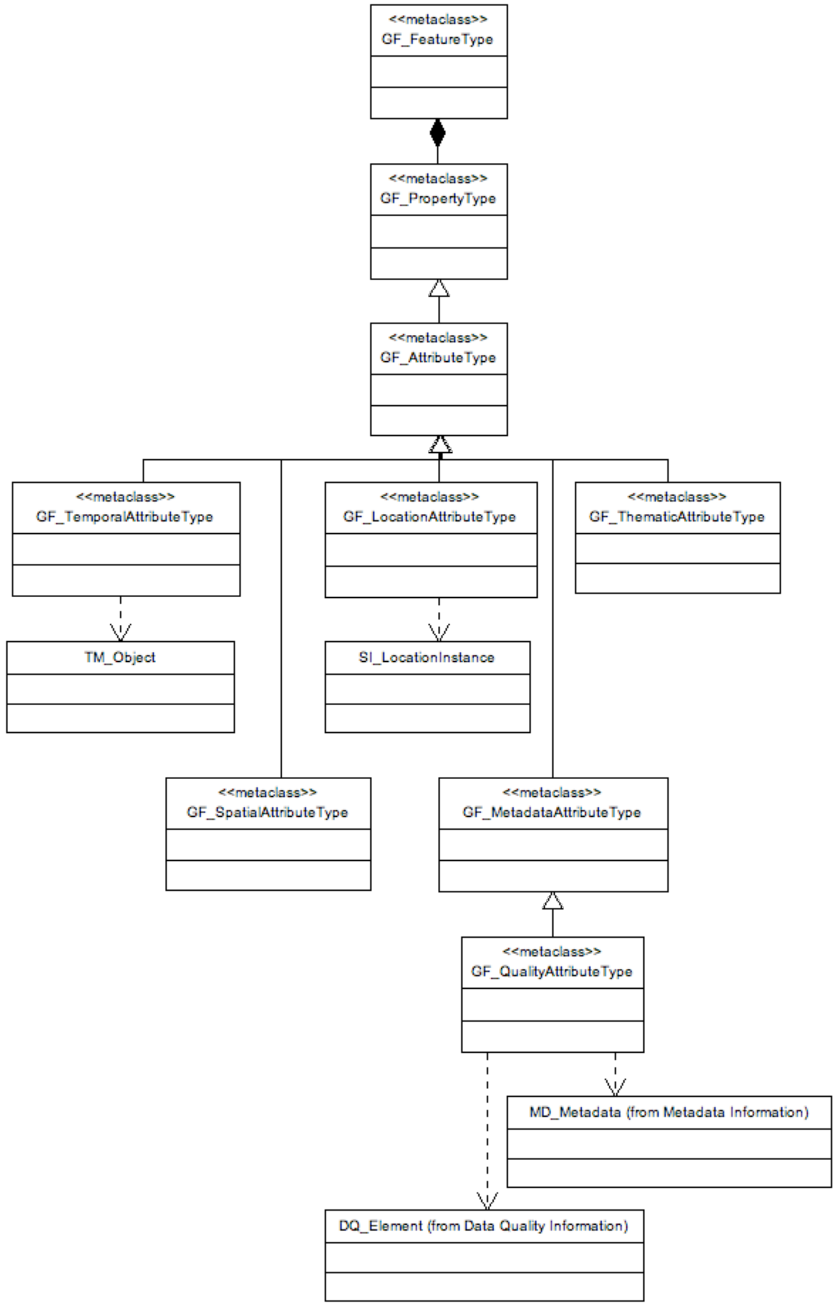


Figure 2. Attributes of feature types (adapted from ISO 19109)

3. Feature Metadata Implementation

To illustrate how this conceptual model translates into a practical ISO-compliant treatment for embedded feature-level metadata, an existing published standard was identified and transformed. Providing a sparse approach for feature depiction, the USDA Forest Service's feature metadata standard (summarised in Table 1) was therefore deemed appropriate.

Attribute	Data Type	Description
REV_DATE	Date	Date of feature creation or revision
DATA_SOURCE	Character (2)	Source of feature
ACCURACY	Number (6,2)	Feature accuracy measured in dataset units of measure

Table 1. Attributes of feature types used in the USDA Forest Service feature metadata standard[1]

3.1 REV_DATE

Denoting the date instance of feature creation or update, the ISO equivalent of REV_DATE may be arrived at through traversing the schema illustrated in Figure 3 (as are DATA_SOURCE and ACCURACY below). Two candidate entities are presented: DQ_DataQuality>LI_Lineage and MD_Identification>MD_DataIdentification. The latter is defined as containing information relevant for data identification and so may be dismissed; the former is chosen as it is formally defined as supporting information regarding the “events or source data used in constructing the data” (ISO 19115, clause B.2.4.2.1).

Due to the ambiguity of the REV_DATE definition (creation OR revision), a decision must be made between which specialised LI_Lineage entity to adopt: LI_Source depicts data creation information, LI_ProcessStep the data transformation and maintenance details. LI_ProcessStep is consequently selected due to probability of data being revised, as indicated by the source standard’s attribute name. The entity’s *dateTime* property is thus identified as the ISO equivalent, a field which may more appropriately be implemented using its short (more specific) name *stepDateTm*.

3.2 DATA_SOURCE

LI_ProcessStep entity is also used to define the DATA_SOURCE field; its *description* property (short name *stepDesc*) is chosen as it specifically provides for a narrative of the data creation process. And as this is constrained by a non-prescriptive free text domain, incorporation of existing dictionaries are permitted (such as that defined within the example USDA standard).

3.3 ACCURACY

Assessments of a data object’s quality are documented in ISO 19115 via the subclasses of the DQ_Element entity. The positional accuracy field as defined in the USDA standard is consequently represented by the DQ_AbsoluteExternalPositionalAccuracy specialisation of DQ_DataQuality>DQ_Element.

[1] http://www.fs.fed.us/gac/metadata/feature_level.html

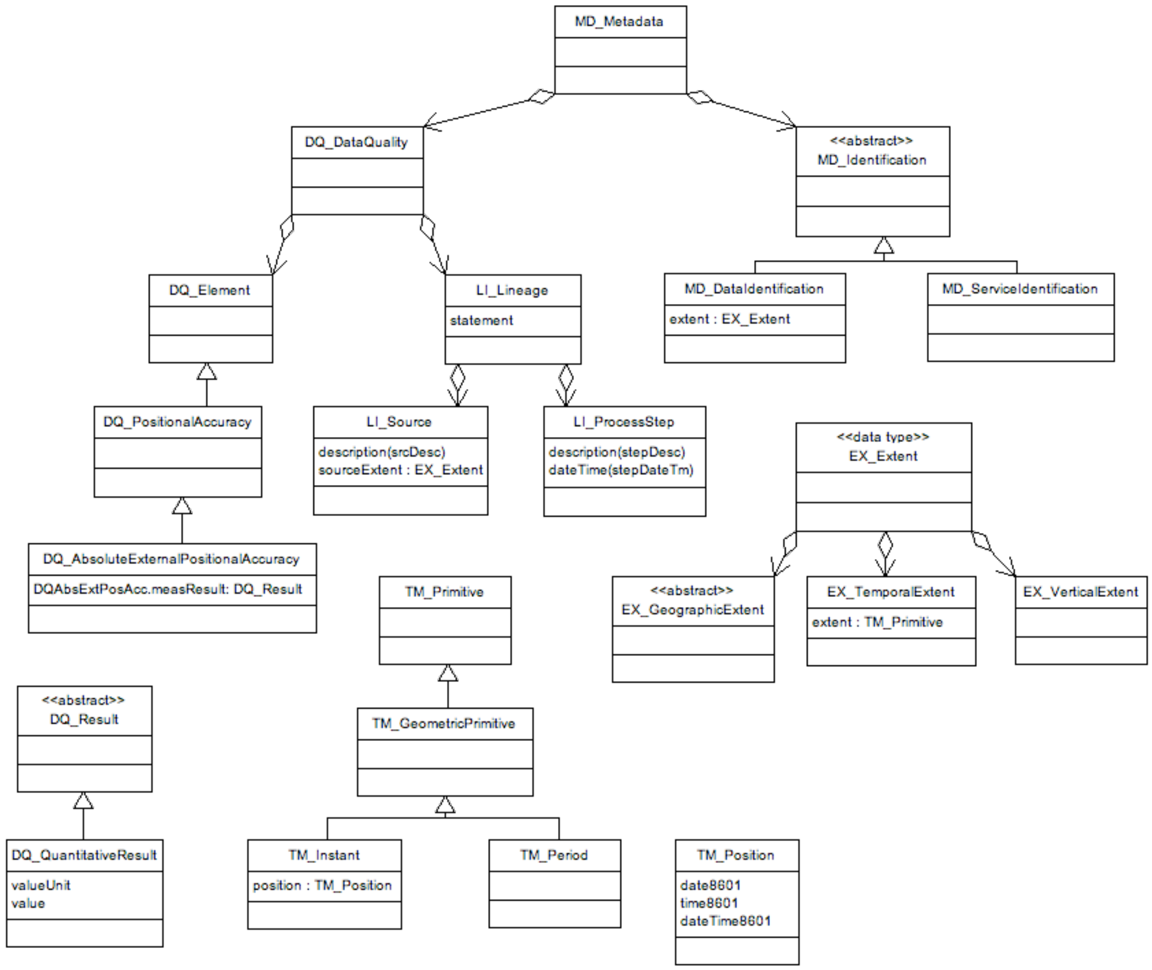


Figure 3. ISO 19115 metadata objects used in the definition of feature metadata

3.4 Character Encoding

While strict ISO compliance requires that the character encoding for each feature type be respected (and are consequently included herein), decisions as to whether this degree of adherence is desirable will depend upon the application. Disambiguation may be necessary where feature provenance is unclear (such as in some web applications) whereas a less stringent encoding may suffice for quality tracking at , for example, organisational level.

Attribute	Data Type	Character encoding
stepDateTm	DateTime	dateTime (ISO 8601)
stepDesc	Character String	Free text
DQAbsExtPosAcc	Number	Record (ISO 19103)

Table 2. ISO equivalents of USDA feature types

4. FLM Applications

Current and prospective uses for embedded metadata at the feature level are various. With the increasing volumes of information being produced and communicated, the benefits of being able to update only those portions of a dataset which have changed are clear – as illustrated by the Ordnance Survey's Change Only Updates service. Other applications which have leveraged embedded metadata have focussed on quality control and the representation of data 'fuzziness' (Kennedy, 2000), feature-time series analyses (Goodall *et al*, 2004) and (attribute) data mining (Merrett, 2002).

The following meanwhile outlines some preliminary musings on potential areas of investigation surrounding feature level metadata. Topics for consideration within each include the degree of schema detail, the encoding mechanism (whether embedded or associated), the method for querying and retrieval, proprietary versus open source implementations, among others.

4.1 Metadata generation

Whether yielding elements via type definitions, or allowing for more accurate assessments of data update events, feature level metadata can play a potential role in automated processes aimed at automatically documenting geospatial assets at higher degrees of resolution. While potential contribution may well prove modest, coupling feature metadata analysis with other metadata authoring processes will have a cumulative impact.

4.2 Asset management and visualisation

Real-time visualisation of data resources for activities such as quality control, workflow allocation and productivity surveillance is often tied to proprietary systems requiring specific clients and data formats. Coupling feature metadata with open source web service technologies provide the potential for overcoming restrictive approaches and enable local and remote connectivity.

4.3 Feature semantics

Unlocking the semantics inherent in features and their metadata may provide for a different approach in verifying data validity. Such an approach could involve associating feature types with an ontology encoded with rules, such as defining the geographic occurrence of certain objects, or the permitted surveying techniques for a given category of feature. Processed using computerised reasoning approaches, inconsistencies can thus be identified and addressed.

5. References

GOODALL, J.L., D.R. MAIDMENT, and J. SORENSON. 2004. Representation of Spatial and Temporal Data in ArcGIS, AWRA GIS and Water Resources III Conference, Nashville, TN.

ISO 2005. BS EN ISO 19115:2005. Geographic information - Metadata.

ISO 2005. BS EN ISO 19115:2005. Geographic information - Rules for Application Schema.

KENNEDY, M. 2000. Embedded metadata - quality control with the dot probability paradigm and ArcQC. Proceedings of the Twentieth Annual ESRI User Conference.

MERRETT, T.H., 2002, Attribute Metadata for Relational OLAP and Data Mining. Lecture Notes in Computer Science, 2397, pp. 97-118.

Biography

James Batcheller is currently in his second year as a PhD student in the School of GeoSciences, University of Edinburgh. He has a background in IT and Environmental Sciences and has recently returned to his studies after a number of years in the GIS industry.

Femke Reitsma is Director of the MSc in GIS course in the School of GeoSciences, University of Edinburgh. Her my research interests include modelling processes, spatio-temporal things, data models, geosemantics, complexity theory, agent based modelling, systems theory, applied process philosophy, and ontologies.

Bruce Gittings is a member of the Edinburgh Earth Observatory, University of Edinburgh involved in GIS teaching and research. His particular interests include database management and web-based GIS.

Automating geospatial metadata using ESRI's ArcGIS and Microsoft's .NET

James K Batcheller, Robert I Dunfey, Femke Reitsma and Bruce M Gittings
Edinburgh Earth Observatory - School of GeoSciences - University of Edinburgh
Drummond Street - Edinburgh EH8 9XP - Scotland, U.K
Tel. + 44 131 650 2565 Fax + 44 131 650 2524
{jk.batch|femke.reitsma|bruce}@ed.ac.uk

INTRODUCTION

Geospatial metadata has long played an important role in the management and location of geospatial datasets (Kim, 1999; Tsou, 2002; Limbach et al., 2004). Often employed by institutions to organise, maintain and document their geographic resources internally, metadata may also provide a vehicle for exposing marketable data assets externally when contributed to on-line geospatial exchange initiatives such as the UK's public sector metadata service *Gigateway* (Batcheller and Gittings, 2006). In spite of the numerous benefits afforded, obstacles to the production of such metadata are numerous (Mathys 2004). Perceptions of it being a tedious yet arduous task, coupled with an assignment of low priority even where the advantages are appreciated all too often result in what may be referred to as the *metadata bottleneck* (Liddy *et al.*, 2002). The current work proposes an approach aimed at reducing the effort associated with geospatial metadata generation through the customisation of a proprietary GIS. By coupling data preparation, management and documentation approaches with such a bespoke application, it is intended to mitigate impediments to geospatial metadata generation whilst promoting a system of data administration that safeguards the data it supports. Geospatial metadata has long been advocated to facilitate the management of data collections; the current approach takes this one step further, using metadata standard elements to coordinate data filing and in the process, contribute to metadata production.

APPROACH

The prototype was designed to integrate a systematic data management model with data initialisation and documentation processes, the aim being to conflate the component workflows whilst facilitating the automatic creation of appropriate metadata. Developing the tool within an existing GIS suite complete with metadata support offers a means by which data creation and editing can be bound more closely to that of its metadata, mitigating the data – metadata disconnect and minimising the risk of inconsistency. ESRI's ArcGIS was chosen due to its extensible ArcObjects-based architecture of modular programming components with which software can be rapidly deployed. Further, by providing a "framework for the implementation of a custom metadata environment" (Vermeij, 2001), its ArcCatalog application offers an extensive pre-existing toolkit with which to develop. The platform for development used was Microsoft's .NET, chosen both for its support for solution extensibility and in its tight integration with XML technologies (Stephens and Hochgurtel, 2002). The personal geodatabase was selected as the test data storage model due its positioning between (legacy) hybrid single-user file-based data stores and integrated multi-user database strategies (Batcheller *et al.*, 2007). A Qualified Dublin Core profile with geospatial refinements was also defined, providing a concise set of twenty-three elements against which the prototype could be evaluated (Table 1).

Core Element Name	Element refinement	Description
Title	-	Title
	Alternative	Alternative title
Description	Abstract	A brief narrative summary of the dataset
Language		Language
Subject	Keywords	Main dataset theme(s)
Date	Created	Date of creation
	Modified	Last date of update
	Period.name	Name of a specific interval. Used here to define frequency of dataset update
Creator		Originating person / organisation
Publisher		Distributing person / organisation
Contributor		Contributing person / organisation
Format		Digital manifestation of resource
Type	Dataset	Nature of content
Rights	Access Rights	Access restrictions
Coverage	Spatial.Box.name	Name of geographic extent of dataset
	Spatial.Box.projection	Spatial reference system of dataset
	Spatial.Box.northlimit	Limits of dataset extent in coordinates
	Spatial.Box.eastlimit	
	Spatial.Box.southlimit	
	Spatial.Box.westlimit	
Identifier		Online linkage to dataset
Relation		A reference to a related resource
Source		A reference to a resource from which the present resource is derived.

Table 1. Qualified Dublin Core element set used to document the test dataset and evaluate the metadata tool. Fifteen core elements are qualified by an additional eight element refinements, providing twenty three fields in total.

TOOL EXECUTION

On selecting a dataset within ArcCatalog the prototype is initiated via a standard button interface, presenting the user with a form on which metadata elements may be edited and which also functions as the principal mechanism through which the utility is controlled. Any pre-existing metadata items held with the dataset are immediately collected on form load; elements may be manually edited, added if empty or selected for overwrite using the prototype's routines. In addition, all operations may be selected to run simultaneously, individually or in various combinations, allowing full control over what routines are executed. The operations the prototype performs are illustrated in Figure 1.

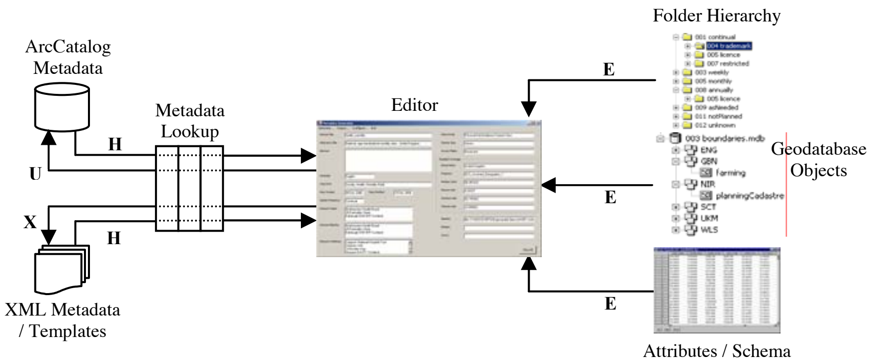


Figure 1. Flow diagram of the operations performed by the current metadata tool, generating elements from a number of separate sources for review on the main editing form.
H: Harvesting; E: Extraction; X: Export; U: Update.

ROUTINES

Element Harvesting

Harvesting routines are run using XPath expressions, defining from whence to retrieve pre-formed metadata from both internal, ArcCatalog-specific XML (stored alongside the dataset in question) as well as from external user-defined XML templates . In the case of the latter, system variables read from the underlying operating system (such as workstation domain and username) can be used to determine the appropriate templates to query. XPath expressions are encoded in a lookup table interpreted by the tool and which may be readily adapted to a variety of metadata conventions as well as used as a fundamental crosswalk for metadata output.

Element Extraction

Custom routines are used to extract further information from the dataset, its data content as well as the dataset's location within a refined folder hierarchy.

Folder & Geodatabase Hierarchy

Metadata entities are used to organise the very data they describe, providing a nomenclature with which datasets may be tagged, categorised and stored. Personal geodatabases, their contents and the folders in which they are held are labelled according to appropriate metadata vocabulary terms by which they may be unambiguously characterised (Table 2), facilitating the logical, hierarchical management of data stores whilst contributing towards the automated compilation of their corresponding metadata records. The hierarchy initially employed by the prototype is illustrated in Figure 2.

Container	Name	ISO Code List
Primary tier	Date Period	19115:MD_MaintenanceFrequencyCode
Secondary tier	Access Rights	19115:MD_RestrictionCode
Personal geodatabase	Subject Keyword	19115:MD_TopicCategoryCode
Feature dataset	Coverage Spatial Box Name	3166-2
Feature class	Subject Keyword	19115:MD_TopicCategoryCode

Table 2. Prototype folder hierarchy in which datasets are tagged and filed, employing entities from code lists (vocabularies) of commonly used ISO standards.

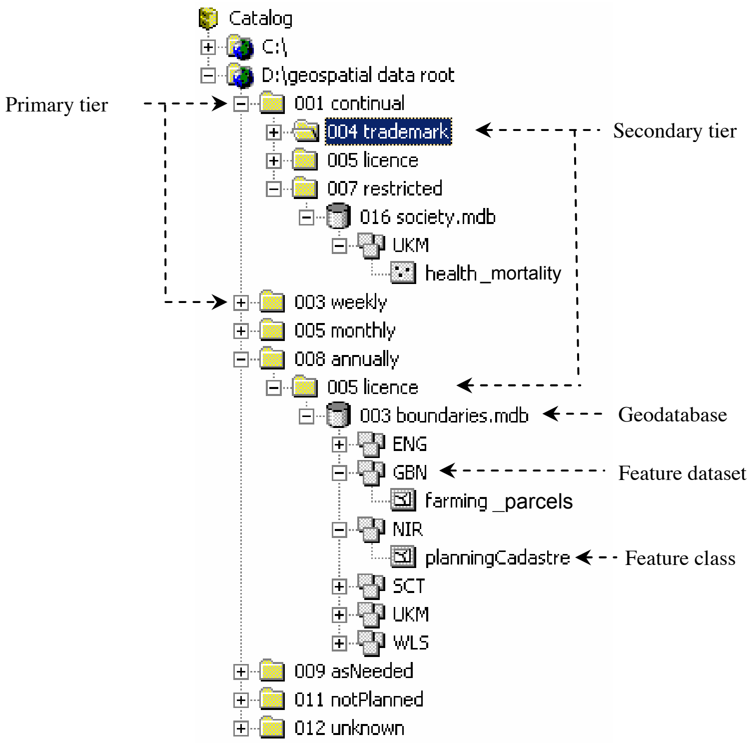
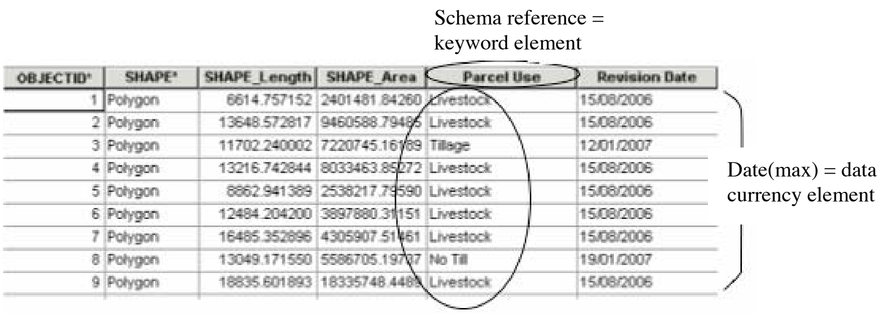


Figure 2. Illustration of current data storage hierarchy which yields five metadata values. Container tags are based on specific metadata entity terms provided above in Table 2.

Data and Dataset

Additional dataset properties not formally treated as items of metadata within ArcCatalog but which are nevertheless programmatically accessible may also be extracted. The current Dublin Core profile's Alternative Title is thus generated; other exploitable properties include spatial resolution and certain elevation values.

Feature attribute instances and attribute schemas may also be leveraged to contribute towards metadata production. Providing predictable feature catalogue-based schemas[1] are adhered to, metadata items may be extracted through the use of indexing techniques, functions performed against the attribute values of a specific field or by direct referencing of feature type definitions (Figure 3).



OBJECTID*	SHAPE*	SHAPE_Length	SHAPE_Area	Parcel Use	Revision Date
1	Polygon	6614.757152	2401481.84260	Livestock	15/08/2006
2	Polygon	13648.572817	9460588.79485	Livestock	15/08/2006
3	Polygon	11702.240002	7220745.16169	Tillage	12/01/2007
4	Polygon	13216.742844	8033463.85272	Livestock	15/08/2006
5	Polygon	6862.941389	2538217.79590	Livestock	15/08/2006
6	Polygon	12484.204200	3897880.31151	Livestock	15/08/2006
7	Polygon	16485.352896	4305907.51461	Livestock	15/08/2006
8	Polygon	13049.171550	5586705.19707	No Till	19/01/2007
9	Polygon	18835.601893	18335748.4489	Livestock	15/08/2006

Figure 3. Elements are extracted by referencing a predictable attribute schema; specific attribute fields may yield elements via indexing or custom functions.

EDITING, OUTPUT AND VALIDATION

The tool centres on a form interface through which routines are initiated and metadata items are edited. Extraction and harvesting routines can either be performed automatically or individually executed once the form has loaded; similarly, elements can be interactively deselected to prevent being overwritten. On form completion, records may be output to ArcCatalog-native format for storage alongside the data, as well as exported to XML files conforming to those standards depicted in the metadata crosswalk file. Further validation routines may be incorporated via on-form spell-checking and domain lookups or may involve more stringent XML schema-based validation supported by the Microsoft XML Core Services which accompany the .NET platform.

RESULTS

Of the total twenty three metadata standard elements outlined above, twenty were completed using the proposed approach (Table 3); the compound element "keyword" comprising of four sub-elements retrieved through the various extraction methods.

[1] The ISO 19109:2005 Geographic Information – Rules for Application Schema standard for example permits the definition of conceptual data models to define the logical structure of a particular application's data, commonly instantiated using feature catalogues that define permissable feature types

Routine	Element (abridged)
Harvesting - pre-formed metadata	Title; Language; Date Created; Format; Dataset Type; Projection; Spatial Box Coordinates; Identifier – 11 total
Harvesting – external templates	Creator; Publisher; Contributor – 3 total
Element Extraction – hierarchy	Date Period; Access Rights; Spatial Box Name; Keyword (x2); - 3 ½ total
Element Extraction – Dataset	Alternative Title – 1 total
Element Extraction – Data	Date Modified; Keyword (x2) 1 ½ total

Table 3. Breakdown of metadata items retrieved and the corresponding routines used. Qualified Dublin Core elements not completed include Abstract, Relation and Source.

DISCUSSION

It could be argued that what is presented here is not so much the automatic generation of metadata but the transfer of effort from metadata authoring to data preparation and management. While this is certainly, but not exclusively, the case, it is put forward as a sound model for metadata management as it promotes a considered approach to data storage as well as sound data preparation. Furthermore, it enables the release of authoring resources which may be redirected towards more intellectually challenging metadata tasks such as descriptive metadata creation and quality control – a conspicuous advantages in cases where data documenters and data authors or managers are distinct. It can also serve to safeguard metadata quality – contingent on appropriate dataset categorisation and data preparation – as the majority of elements are no longer entered manually and susceptible to human error. And while the data storage strategy proposed herein may be quite reasonably viewed as contrived; the opinion held here is that data management, by definition, should adhere to a predictable, formal schema to best allow data categorisation and subsequent retrieval. In all, it is contended that the current approach has the potential to offer a significant net saving of time for applications reliant on the production of metadata, despite the potentially high initial investment.

CONCLUSIONS AND OUTLOOK

The metadata management framework outlined above supports users in reducing the effort involved in documenting data, ensuring that a minimum amount of elements are automatically generated according to relevant metadata standards and best practice. Considering the parallels with the digital library and internet cataloguing arenas, where resource volumes make it “unrealistic to depend on traditional humanly-generated metadata approaches” (Greenberg *et al.*, 2006 p3), efforts to streamline metadata creation though automation begin to take on more importance. The future of generating useful metadata involves increasing computational support to minimise human effort; advances in representing the semantics of metadata may well have particular relevance for automating its collection and exploitation. In conclusion, while the approach presented was one bound to a particular proprietary solution, the objective was not to laud one offering above all others but to highlight the potential contribution a dataset’s ambient computing infrastructure can make in automating the creation of geospatial metadata.

BIBLIOGRAPHY

Batcheller, J. K. and Gittings, B. M., 2006. Avenues for developing the UK's National Geospatial Metadata Service. Proceedings of the GIS Research UK 14th Annual Conference, University of Nottingham, Nottingham, UK, pp.259-262.

Batcheller, J. K., Gittings, B. M. and Dowers, S., 2007. The Performance of Vector Oriented Data Storage Structures in ESRI's ArcGIS. Transactions in GIS 11(1): 47-65.

Greenberg, J., Spurgin, K. and Crystal, A., 2006. Functionalities for automatic metadata generation applications: a survey of metadata experts' opinions. International Journal of Metadata, Semantics and Ontologies 1(1): 3-20.

Kim, T. J., 1999. Metadata for geo-spatial data sharing: A comparative analysis. The Annals of Regional Science 33: 171-181.

Liddy, E.D., Allen, E., Harwell, S., Corieri, S., Yilmazel, O., Ozgencil, N.E., Diekema, A., McCracken, N.J., Silverstein, J. and Sutton, S.A., 2002. Automatic metadata generation and evaluation. Proceedings of the 25th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, Tampere, Finland, pp.401–402.

Limbach, T., Krawczyk, A. and Surowiec, G., 2004. Metadata Lifecycle Management with GIS Context. 10th EC GI & GIS Workshop, ESDI State of the Art, Warsaw, Poland.

Mathys, T. (2004). The Go-Geo! Portal Metadata Initiatives. In: Proceedings of the GIS Research UK 12th Annual Conference, University of East Anglia, Norwich, UK, pp.148-154

Stephens, R., Hochgurtel B., 2002 Visual Basic .NET and XML. John Wiley & Sons Inc, New York. ISBN 047112060X

Tsou, M.-H., 2002. An Operational Metadata Framework for Searching, Indexing, and Retrieving Distributed Geographic Information Services on the Internet. In: Egenhofer, M. and Mark, D. (Eds.) Lecture Notes in Computer Science Vol. 2478, Springer-Verlag, Berlin, pp. 313-332.

Vermeij, B., 2001 Implementing European Metadata Using ArcCatalog - ArcUser Online. http://www.esri.com/news/arcuser/0701/metadata.html. Last accessed: 17th January 2007.

Computers & Geosciences 34 (2008) 387–398

COMPUTERS & GEOSCIENCES

www.elsevier.com/locate/cageo

Automating geospatial metadata generation—An integrated data management and documentation approach

James K. Batcheller*

Edinburgh Earth Observatory, Institute of Geography, School of GeoSciences, University of Edinburgh, Drummond Street, Edinburgh EH8 9XP, UK

Received 15 January 2007; received in revised form 30 March 2007; accepted 4 April 2007

Abstract

Geospatial metadata have long played an important role in the management of geospatial datasets. Often employed by institutions to organise, maintain and document their geographic resources internally, metadata may also provide a vehicle for exposing marketable data assets externally when contributed to on-line geospatial exchange initiatives. In spite of the numerous benefits it affords, obstacles to the production of such geospatial surrogates are numerous. The current work proposes an approach aimed at reducing the effort associated with geospatial metadata generation through the customisation of a proprietary Geographical Information System (GIS). By coupling data preparation, management and documentation approaches with such a bespoke application, it is intended to mitigate impediments to geospatial metadata generation whilst promoting a system of data administration that safeguards the data it supports. The current prototype, implementing an extended Dublin Core geospatial profile of 23 elements, was capable of generating a total of 20 basic metadata entries. While the findings do not suggest a dispensability of human mediation in the authoring process, they do support the view that a dataset's ambient computing infrastructure has the potential to play a significant role in automating the creation of geospatial metadata.

Keywords: Geospatial metadata; Metadata authoring; Metadata generation; Data management; Data documentation

1. Introduction

Since their appearance in the latter half of the twentieth century, the proliferation of Geographical Information Systems (GIS), their applications and related technologies has continued apace (Goodchild and Haining, 2004). With the more current developments in the realm of web-enabled geospatial services, as well as the emergence of popular Geographical Exploration Systems (GES) such as *Google Earth* and Microsoft's *Virtual Earth*, increasing numbers of people continue to be introduced to the possibilities afforded by such technologies. Whether for public, private or academic purposes, demand for geographical information (GI) has in addition increased several-fold in recent times, with those looking to procure data turning to the exploration of existing data pools, commissioning the collection of new data, or resorting to producing their own. Efforts to meet this demand contribute to the explosion of data currently available,

*Tel.: +44 131 6502558; fax: +44 131 6502524.
E-mail address: jk.batch@ed.ac.uk

0098-3004/$ - see front matter © 2007 Elsevier Ltd. All rights reserved.
doi:10.1016/j.cageo.2007.04.001

introducing further problems relating to the management of quite often voluminous data holdings, and how such assets can be successfully exploited.[1]

As more data and information is produced, the more vital approaches become for managing and locating such resources (Göbel and Lutze, 1998); the role geospatial metadata assumes here has been widely acknowledged[2] (Kim, 1999; Tsou, 2002; Limbach et al., 2004). Apart from providing a means of documenting a dataset's key statistics such as its quality, appropriateness, currency or area of coverage, metadata can supply information on the availability of the data it describes, how it may be accessed and exchanged; it contributes towards data management efforts by helping to organise, maintain and locate data resources; when collated into catalogues, metadata collections can be indexed for rapid query, contributed to data clearinghouses or similar data exchange initiatives where they can be used to externally expose marketable data assets; it aids in the coordination of data procurement efforts by raising the awareness of extant datasets, thereby avoiding duplication of effort, redundant storage and obscuring search results.

Further incentives for its use arise when the implications of neglecting metadata entirely are considered. Some claim that the cost of not creating metadata can outweigh that of authoring it, citing concerns associated with employee turnover, data redundancy, conflicts and inappropriate decision-making.[3] Others meanwhile go so far as to argue that data is rendered useless in the absence of any metadata (Qi et al., 2004). Despite the arguments, obstacles to the adoption of metadata practices remain. Many view its generation as monotonous and time consuming, a labour-intensive process which is a major undertaking in itself (Guptill, 1999; West Jr. and Hess, 2002), resulting in a pervasive outlook which shuns metadata creation (Mathys, 2004). Streamlining conventional authoring processes, and thereby conserving associated resources, would mitigate the barriers to data documentation.

The negative perceptions of metadata practices can persist even once they have been adopted, often with harmful consequences for the quality of output. Even where its value is recognised, data documentation commonly takes low priority in relation to other activities, reduced to being seen as "a necessary evil."[4] And as conventional geospatial dataset documentation remains a largely manual process, it tends not only to be tedious when finally undertaken, but also error prone (Leiden et al., 2001). Considering that large volumes of data currently on offer emanate from those not traditionally considered to be geospatial data producers,[5] questions arise as to whether the accompanying metadata (when present) consistently reflects that which it purports to document.

The current work proposes an approach aimed at reducing the effort associated with geospatial metadata generation. Further, by combining data preparation, filing and documentation workflows within a combined framework, barriers to the creation of geospatial metadata can potentially be lowered while simultaneously enforcing a system of data organisation designed to safeguard such assets. Regardless of application domain, it is contended that facilitating the accelerated location, retrieval and interpretation of an organisation's data holdings thought the use of metadata can serve to realise the potential of (frequently underexploited) geospatial resources. The paper is structured to provide a review of previous work and leads to the details of the proposed framework; the findings are discussed thereafter, followed by the conclusions drawn.

2. Related work

2.1. Digital library and information science community

Given the rapid and continual growth of accessible digital resources observed since the advent of the World Wide Web, it is unsurprising that efforts to facilitate effective information location, navigation and retrieval through resource documentation have followed. The digital library and Internet cataloguing arenas have hosted a number of

[1]Yuan, M., Buttenfield, B., Gahegan, M.N., and Miller, H., 2001. Geospatial Data Mining and Knowledge Discovery. http://ags.ou.edu/~myuan/papers/mining.pdf.

[2]Kacmar, C., Jue, D., Stage, D., and Koontz, C., 1995. Automatic Creation and Maintenance of an Organizational Spatial Metadata and Document Digital Library. http://csdl.tamu.edu/DL95/papers/kacmar/kacmar.html.

[3]Deng, Y., 2002. The Metadata Architecture for Data Management in Web-based Choropleth Maps. http://www.cs.umd.edu/projects/hcil/census/JavaProto/metadata.pdf.

[4]Vermeij, B., 2001. Implementing European Metadata Using ArcCatalog—ArcUser Online. http://www.esri.com/news/arcuser/0701/metadata.html.

[5]Schweitzer, P.N., 1998. GIS and Metadata—Putting Metadata in Plain Language. www.geoplace.com.

research initiatives investigating automated metadata generation, motivated by the view that it is "unrealistic to depend on traditional humanly generated metadata approaches" when considering the volumes of resources involved (Greenberg et al., 2006).

Greenberg (2003) elaborates a framework for metadata generation for online content, noting the part standards play in guiding metadata authoring in addition to the roles of human and computing resources. Automated practices therein are categorised into those which employ resource content indexing i.e. are not predicated on the presence of recognised metadata elements, and those employed by commercial search engines, whether using preformed metadata or that produced at run-time. Liddy et al. (2001) suggest that such automated techniques can produce reasonable results in certain circumstances; Anderson and Pérez-Carballo (2001) maintain that automated methods tend to be more efficient, consistent and inexpensive than human ones. Whatever the proposed method, most agree that automated and manual approaches combined promise the most in producing quality resource documentation (Craven, 2001; Greenberg, 2004).

2.2. The geospatial domain

2.2.1. Geospatial data

The very nature of geospatial data dictates a somewhat different approach than those mentioned above. GI data tend to be both highly structured and manifested in a variety of forms, characterised by the presence of some treatment for geometry. Storage techniques vary—even within proprietary systems—from hybrid models that store spatial and attribute information separately across different files to integrated strategies employing relational databases (Batcheller et al., 2007). Most geospatial storage formats therefore do not lend themselves to the same probing operations as used with textual resources given their content's relative lack of accessibility.

The lack of sophisticated support for metadata within pioneering geospatial storage strategies meant that any important information not encoded within a dataset needed to be documented elsewhere. Externalising metadata in discrete text files not only bypassed the need for opening often large documents in their host applications when certain dataset attributes were sought; it would also enable the use of existing indexing and cataloguing techniques for data location and management previously mentioned. Authoring tools consisted of common text editors with metadata often recorded in ad hoc or institution-specific conventions with few common guidelines and little provision for interoperability.

2.2.2. Geospatial metadata standards

More recently, geospatial data documentation efforts have been underpinned by the use of standards, viewed as important keys in facilitating metadata exchange, interpretation by individuals and manipulation by machines. Rising from the initial foundations laid by early data standard initiatives (Moellering, 1992), geospatial metadata standards, like their mainstream counterparts, aim to define metadata content and structure. Content standards describe a "common set of terminology and definitions for the documentation of digital (geospatial) data" (FGDC, 1998), while metadata encoding standards, commonly implemented using XML Document Type Definitions (DTD) and XML Schemas, outline how content is manifested digitally. Used in tandem, these standards in many ways simplify metadata generation, offering a template for content as well as providing guidelines for permitted input.

Considering the detail to which common geospatial metadata conventions are elaborated however, standards can also have a simultaneously detrimental effect, complicating metadata generation and potentially undermining implementation initiatives (Tsou, 2002). In the United States, the FGDC's[6] Content Standard for Digital Geospatial Metadata (CSDGM) of 1998 outlines a standard containing over 300 data and compound elements (FGDC, 1998). The recently ratified ISO 19115: 2005 standard for Geographic Information meanwhile details a metadata element set of over 400 (ISO, 2005). Clearly the benefits afforded by full compliance to either standard will be significantly outweighed by the resources necessary to achieve it.

Metadata standard profiles have consequently arisen for a variety of application domains, the creation of which may themselves be guided by formalised standards such as ISO 19106:2006 Geographic Information—Profiles (ISO, 2006). Essentially subsets of a given metadata convention, profiles define a limited set of elements designed for a specialised purpose whilst simultaneously

[6]The Federal Geographic Data Committee.

maintaining standard compliance, often simplifying the metadata authoring in the process. Profiles have for instance been developed to enable data location (*discovery metadata*), to help potential users make decisions on a dataset's appropriateness (*exploration metadata*) and to facilitate data utilisation (*exploitation metadata*) (Taylor, 2004). Similarly, region-specific profiles abound, such as the Euro-centric ISO profiles overseen by CEN[7] (Longhorn, 2005). While evidently useful for reducing the effort of documenting data, profiles on their own make only a minor contribution in improving the issues relating to metadata authoring efficiency.

2.2.3. Geospatial metadata creation tools

The adoption of standardised approaches to metadata creation, not least the growing popularity of XML, has made the development of generic tools to aid such practices worthwhile. And with the advent of national and international data exchange initiatives,[8] increasing focus has been lent towards the development of tools that encourage the production of consistent, conformant metadata.

Early stand-alone desktop tools provided the metadata author with an interface for entry, with standards-based output directed to either file or relational database storage schemas (cf. *askGIraffe's* customised Microsoft Access tool (Foy, 2001)). Later versions incorporated on-edit or on-export error trapping—using domain lists, DTD or XML schema validation—as well as providing support for metadata parsing, importation and (rudimentary) conversion. Crafted predominantly in the Java or Visual Basic development environments, such tools are characterised by their independence from the proprietary applications commonly used to create and edit the data to be documented. As such, these metadata editors can also function as viewers, permitting the browsing of key (recorded) dataset themes without the need of a GIS suite (West Jr. and Hess, 2002). Examples of such editors include GIS-tec's *Metadata InGeo EntryTool* (Limbach et al., 2004) and GIgateway's *MetaGenie Desktop* (Batcheller and Gittings, 2006).

Whilst desktop editors may serve to produce metadata for use both within an organisation and beyond, online editors are predominantly used as components of geospatial data clearinghouses and GIS portals as mechanisms for metadata contribution. Used to streamline the submission process, editors such as EDINA's *Go-Geo! Metadata Creator* (Mathys, 2004) and G-portal's *XML Metadata Resource Editor* (Lim et al., 2005) also serve to help reduce metadata redundancy and replication as records are typically edited where they are hosted. In addition, both strategies are often enhanced through the use of context-sensitive help, tool-tips and option lists designed to guide user input and improve metadata quality.

An important trade-off of employing such independent, cross-platform editors is the disconnect of metadata authoring practices from proprietary application workflows. Dataset creation and editing are consequently detached from metadata creation and editing procedures, necessitating diligent update practices involving at minimum two separate applications. Countering this by providing an integrated workflow through which both data and metadata can be maintained will clearly counter this disconnect, thereby minimising the risk of inconsistency.

2.3.4. Geospatial metadata applications

The advantages of leveraging GIS applications to aid data documentation go further than workflow consolidation. Geospatial suites provide largely unhindered access to data stores they support, an important consideration if streamlining metadata authoring through automation is to be achieved. And while more accessible open data formats are not dismissed,[9] the majority of data in production environments are held in proprietary data stores—if published market share figures are to be believed.[10] Furthermore, considering the inclusion of programming kits within most common GIS applications (in addition to their near-uniform support for disparate data formats) development of bespoke tools is greatly facilitated.

Due to the lack of metadata support amongst the forerunning data stores, early GIS-native tools focussed on metadata extraction or derivation techniques, i.e. where dataset attributes are mined and transformed for use as metadata items.

[7] The European Committee for Standardisation/Comité Européen de Normalisation.

[8] Notable examples include the FGDC's National Geospatial Data Clearinghouse in the US and *GIgateway* in the UK.

[9] For example the XML-based Geography Mark-up Language, or GML.

[10] Market research firm IDC estimated the market share figure held by open-source desktop software in 2002 at approximately 3% (GIS Monitor, 12 June 2003—http://www.gismonitor.com/news/newsletter/archive/061203.php).

Approaches commonly involved executing an application script to extract and export dataset information (e.g. projection details and bounding coordinates) to file for subsequent validation. Nevertheless, external text editors were still required for the manual completion of each metadata record produced. A typical example is the FGDCMETA.AML tool written in Arc Macro Language (AML) for ESRI's ArcInfo and designed for use with the FGDC's CSDGM standard.

As the perceived importance of metadata increased, GIS vendors started to introduce enhanced support for metadata both within their software offerings and alongside their data models. Native support for metadata content and schema standards, often manifested as XML, became a standard feature within many software offerings, as did the ability to edit and author metadata in-package. Many dataset properties were now treated as specific metadata items, and as a consequence could now also be harvested directly where before they were derived. In addition, with the near-universal adoption of XML-based technologies and increasing reliance of vendor-specific programming environments on common development platforms such as Sun Microsystem's J2EE and Microsoft's .NET, sophisticated turnkey extensions to existing GIS software now became a real possibility.

3. The present work

The current paper proposes that the efficiency of geospatial metadata generation can be significantly enhanced through proprietary software customisation and considered data preparation. Apart from recognising its position as market leader in the GIS software field (and thereby offering a familiar platform in which to present the current approach) ESRI's ArcGIS 9.1 was chosen due to its extensive, extensible architecture based on modular programming components (ArcObjects) with which software can be rapidly developed. In addition, its ArcCatalog component "provides a framework for the implementation of a custom metadata environment,"[11] and thus presents an existing toolkit with which to build. The development platform employed was Microsoft's .NET, chosen to exploit both the platform's support for solution extensibility, but in particular its tight integration with XML (Stephens and Hochgurtel, 2002).

A prototype was built in Visual Basic .NET and compiled into a dynamic linked library (dll) file which is registered with the ArcGIS application. Although ArcCatalog provides near-uniform access to a range of data storage techniques, a single model was employed to limit the degrees of freedom of the current analysis. The personal geodatabase, an integrated single-user solution based on Microsoft's Access RDBMS technology, was accordingly selected due to the relative ease in which it is configured as well as its positioning between (legacy) hybrid single-user file-based data stores and integrated multi-user database strategies.

The tool is designed to provide an integrated approach to metadata generation, based on a systematic data management structure and facilitating efficient data documentation, metadata validation and basic translation. Being native to ArcCatalog it is bound with the dataset initialisation, configuration and management workflow; it may however be readily retooled for use in ArcMap for applications where binding metadata creation with the data editing and analysis workflow is preferred.

On nomination of a dataset within ArcCatalog the tool is initiated via a standard button interface. The user is presented with a metadata editing form which functions as the principal interface of the tool. Pre-formed metadata items held alongside the dataset are instantaneously harvested on form load; elements may either be overwritten manually, completed if absent or selected for revision using the tool's metadata routines. Routines may be run collectively or individually, allowing full user control over the tool's operations. The operations the tool support (illustrated in Fig. 1) include:

- Harvesting pre-existing metadata elements generated by ArcCatalog.
- Extracting file hierarchy, data and dataset properties and attributes for use as metadata elements.
- Harvesting user-prepared metadata templates.
- Guiding the visual inspection, modification and completion of metadata records through the structured presentation of record fields on an editing form.
- Enabling the importation from and exportation to other standards through the use of a basic metadata crosswalk.

[11]Vermeij, B., 2001. Implementing European Metadata Using ArcCatalog—ArcUser Online http://www.esri.com/news/arcuser/0701/metadata.html.

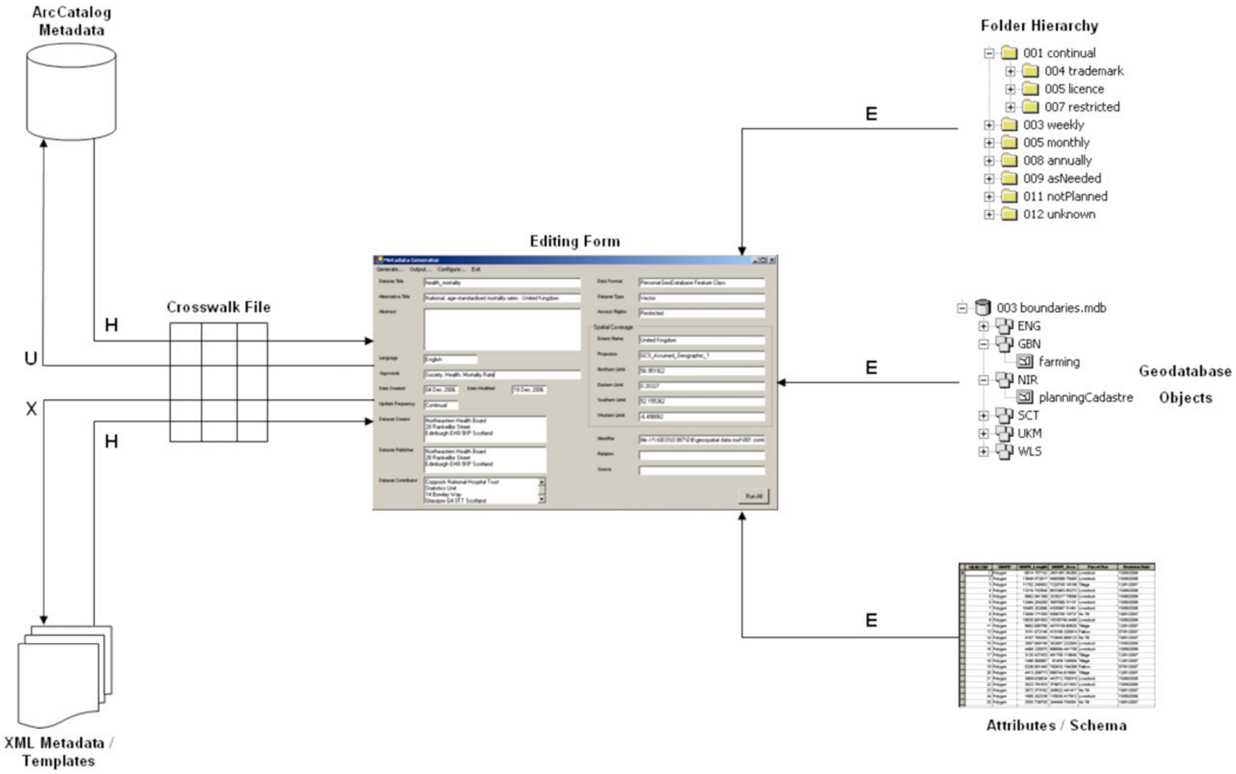


Fig. 1. Flow diagram of current metadata prototype, drawing elements from the various sources. H: harvesting; E: extraction; X: export; U: update.

3.1. Metadata standard

A Qualified Dublin Core profile with geospatial refinements was defined to provide a succinct element set with which to test the metadata prototype (Table 1). Offering a widely adopted convention used for depicting any category of resource, Dublin Core was chosen as it affords a sufficiently sparse and manageable solution within the current context. And as a profile of a well-defined ISO standard, the set's collection of elements is readily mapped to other more detailed geospatial conventions.

3.2. Initial harvesting

Initial harvesting takes advantage of the inherent metadata already collected from registered datasets by ArcCatalog. Stored in ISO-compliant[12] XML alongside the data, prototype routines harvest the required elements contained therein using XPath expressions defined in a custom metadata crosswalk file. Primarily used for cross-mapping conventions as later described, the file details the addresses (in XPath) of elements contained within the dataset's default XML metadata which are retrieved, offering an initial set of fields on which to add. In the current context, seven out of the 23 elements are automatically generated: Title, Language, Date Created, Format, Type, Coverage projection and bounding coordinates as well as Identifier.

3.3. Extraction routines

Custom routines are used to extract further information from the dataset, its data content as well as the dataset's location within a refined folder hierarchy. The latter is based on the premise that efficient data management practices employ logically organised data stores. Here, metadata entities are used to organise the very data they describe, providing a nomenclature with which datasets may be labelled, categorised and filed. In the current configuration, personal geodatabases, their contents and the folders in which they reside are tagged

[12]Specifically, ISO 19115 metadata. Storage in FGDC CSGDSM format is also supported. ArcGIS support for ISO 19139 Geographic information—Metadata—XML schema implementation started with version 9.2.

Table 1
Qualified Dublin Core element set used in the current work

Core element name	Element refinement	Description
Title	–	Title
	Alternative	Alternative title
Description	Abstract	A brief narrative summary
Language		Language
Subject	Keywords	Main theme(s)
Date	Created	Date of creation
	Modified	Last date of update
	Period.name	Name of a specific interval. Used here to define frequency of update
Creator		Originating person/ organisation
Publisher		Distributing person/ organisation
Contributor		Contributing person/ organisation
Format		Digital manifestation of resource
Type	Dataset	Nature of content
Rights	Access Rights	Access restrictions
Coverage	Spatial.Box.name	Name of geographic extent of dataset
	Spatial.Box.projection	Spatial reference system of dataset
	Spatial.Box.northlimit	Limits of dataset extent in coordinates
	Spatial.Box.eastlimit	
	Spatial.Box.southlimit	
	Spatial.Box.westlimit	
Identifier		Online linkage to dataset
Relation		A reference to a related resource
Source		A reference to a resource from which the present resource is derived

Fifteen core elements are qualified by element refinements resulting in a total of 23 fields.

Table 2
Prototype folder hierarchy in which datasets are filed and named

Container	Name	ISO Code List
Primary tier	Date Period	19115:MD_MaintenanceFrequencyCode
Secondary tier	Access Rights	19115:MD_RestrictionCode
Personal geodatabase	Subject Keyword	19115:MD_TopicCategoryCode
Feature dataset	Coverage Spatial Box Name	3166-2
Feature class	Subject Keyword	19115:MD_TopicCategoryCode

Entire code lists need not be replicated within the hierarchy: containers may be created as required on filing new datasets.

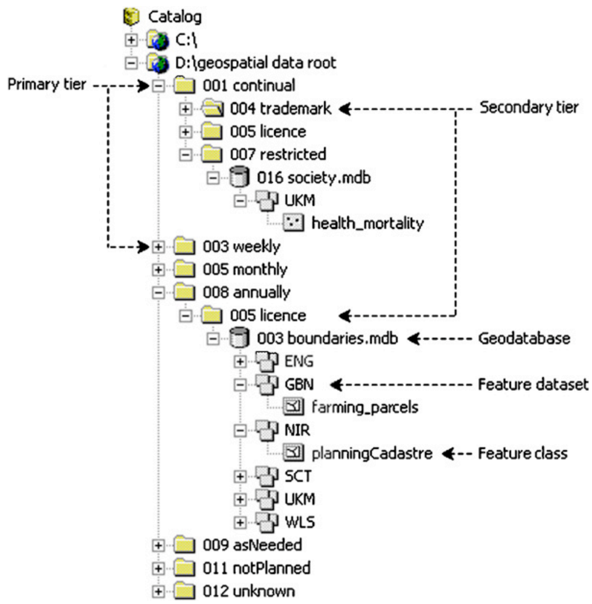


Fig. 2. Illustration of prototype data storage hierarchy, yielding five metadata elements. Container tags are assigned on the basis of specific metadata entity code lists detailed in Table 2.

according to specific metadata vocabulary terms by which they are unambiguously characterised, facilitating dataset management while contributing to automated metadata record compilation.

The test scenario comprised of a three-tiered folder hierarchy—a root directory, a primary tier and a secondary tier. Each tier beneath the root denotes a specific metadata element, holding containers labelled using code lists of commonly used ISO standards (Table 2). Personal geodatabases are similarly tagged and stored in a location within the hierarchy which best reflects the attributes of the data within. Personal geodatabase constituents are likewise managed, with appropriate code lists defining how collections of geographic features (feature classes) and their aggregations (feature datasets) are annotated. The test hierarchy is illustrated in Fig. 2.

The test hierarchy illustrates how the elements of the adopted metadata standard may be used to coordinate dataset storage, and later contribute to metadata record creation. The choice of element for a particular tier will depend on the application domain; Date Period for instance was chosen as primary in the prototype hierarchy to reflect a scenario whereby datasets are organised and filed first by the frequency they need to be updated. Folders (containers) residing on this tier are tagged using the domain codes and names—defined in the ISO 19115:MD_MaintenanceFrequencyCode code list—appropriate for the data to be contained therein. Subsequent tiers and geodatabase objects are tagged in a similar fashion. For metadata contribution, extraction routines read each tag in a given dataset's path; domain code—name pairs may be used in combination or may be parsed as needed.

The approach represents a plausible data management protocol which may be readily adapted to different application domains. The entire hierarchy or geodatabase configuration need not be recreated for effective contribution to generating metadata; indeed, a subset of each code list may be preferred, with containers created only when needed to organise incoming datasets. The contribution of this storage classification strategy is clearly bound to the number of tiers in the combined folder and geodatabase hierarchy. Currently, three simple elements (Date Period, Access Rights and Spatial Box Name) and one compound element (depicting two Subject Keywords) are derived.

The method for extracting metadata elements from a dataset meanwhile presupposes that they have been comprehensively compiled. Additional dataset properties, whether supplied by the author or calculated in the process of registering geographic objects, present the opportunity for extending what can be extracted from a data store. Held alongside the data in a similar manner as spatial referencing information such as projection, coordinate system etc, details may be mined using custom code and transformed into usable metadata elements. Currently a single element—Alternative (title)—is created using this extraction method; scope remains to retrieve other elements not currently treated formally as metadata by the program for more detailed standards (including for instance spatial resolution and certain vertical extent attributes).

The routines which extract elements from the data may depend upon a formal attribute schema, or may allow for a relative lack of structure. Indexing frequently occurring textual attributes may for instance serve to extract values which can be adopted as keyword elements; retrieving feature type definitions may nevertheless suffice if a feature catalogue-based schema[13] is adhered to. Date fields may similarly be queried indiscriminately to yield potential maximum and minimum values for a date range of use element, or they may be referenced directly in the event that a predictable data standard is employed. In the present work, a Revision Date element is derived from the contents of a specific field denoting the date of update of each individual feature.

3.4. Template harvesting

In a method adopted by many geospatial metadata editors, reusable content may be stored in XML files for harvesting during metadata production. The approach is extended herein through the association of variables managed by the underlying operating system with these pre-prepared templates. Details of a dataset's Creator, Publisher or Contributor for the current Dublin Core profile can for instance be automatically incorporated within each metadata record on the basis of variables such as the current username or the domain of the user's workstation. XML template constituents are again addressed using XPath expressions and retrieved in a similar manner as the method for inherent metadata above.

3.5. Metadata editing interface

The tool centres on a form interface through which routines are initiated and metadata items are edited, with the form's fields corresponding to the content of the chosen metadata standard (Fig. 3). Inherent metadata elements retrieved through initial harvesting are used to pre-populate the form, such elements being the most up-to-date and which are typically not edited. Extraction and harvesting routines can either be performed during initial pre-population, or manually executed once the form has loaded on-screen. Here, standard elements (for which metadata generating procedures may be applied) are selected in either their entirety or in any specified combination prior to initiation. Metadata items retrieved may be edited and additional ones supplied in an interface complete with similar content guiding mechanisms as those of the stand-alone editors mentioned previously.

[13]ISO 19109:2005 Geographic Information—Rules for Application Schema standard for instance allows for the definition of conceptual data models which define the logical structure of an application's data, commonly instantiated by feature types defined in feature catalogues.

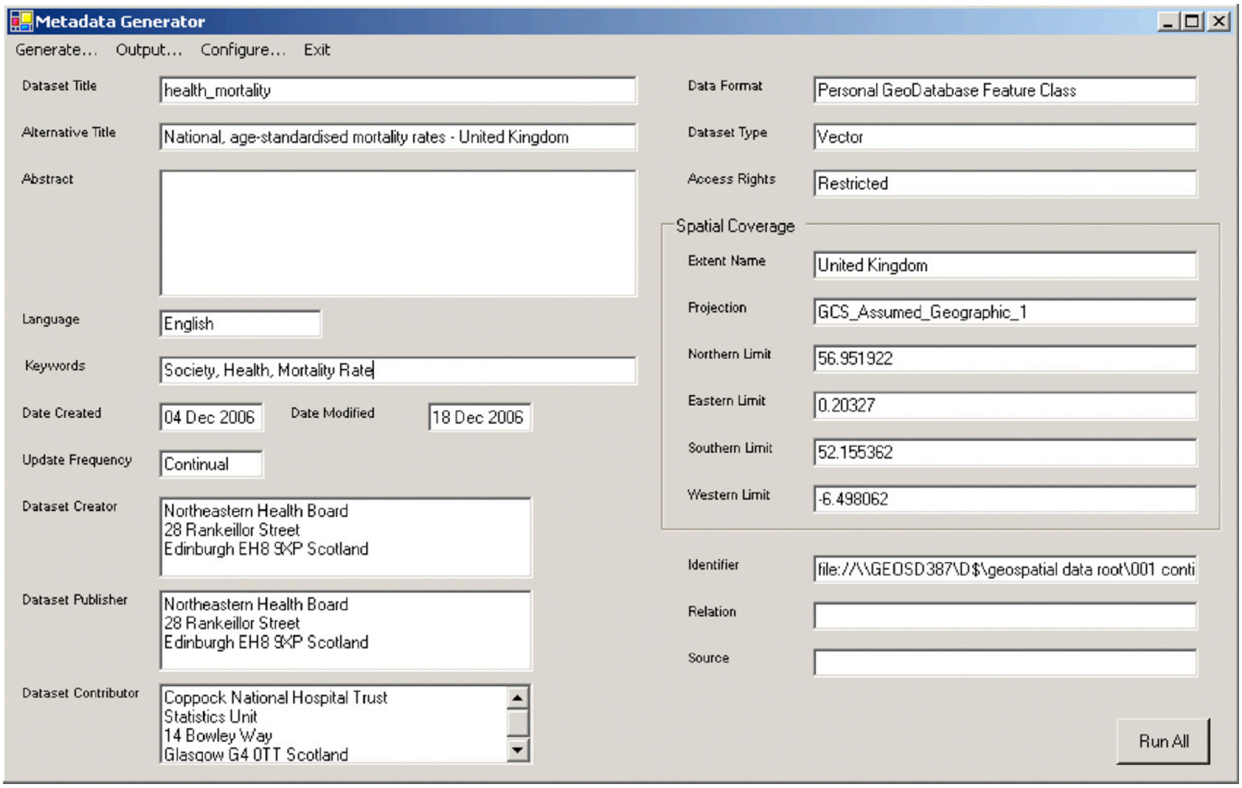


Fig. 3. Prototype's central interface. Routines may be run selectively via the Generate/Output menu items or collectively via the Run All button. Routines and elements may also be selected/deselected (via the Configure menu), enabling mediator customisation of the Run All operation set.

3.6. Record validation

The method for validating completed metadata records will depend upon the application for which they are intended. Records destined for publication on a geospatial data clearinghouse service may demand strict quality control to ensure that both the XML output is well-formed and the content values are within allowable ranges; applications with less stringent quality demands may simply require rudimentary content validation. XML Schema validation in the first instance may be facilitated with the support of the Microsoft XML Core Services (MSXML) which accompany the .NET platform. Metadata records produced to meet a specific standard are output to XML where they may be tested for compliance (using methods provided by MSXML) against a corresponding XSD schema registered with the tool. As an alternative to schema validation, metadata editor fields may be verified prior to export to file using integrated spell-checking, domain value look-ups and other integrity measures such as verification of mandatory field completion.

3.7. Metadata output

One of the key components of the tool is the standard mapping or crosswalk file. Not only used to support the aforementioned metadata harvesting techniques, the file also provides a means of cross-referencing metadata standard elements. Each row in the mapping table denotes a metadata element; columns denote each specified metadata standard. Field values are in the form of XPath references which are used to read from and write to XML metadata. Once the metadata editing form is complete, elements can be written back to the ArcCatalog-native metadata for association with the active dataset or exported to external XML files according to the standard(s) included in the crosswalk file.

4. Results

The prototype offers a reasonable saving in effort for the metadata producer, albeit measured in the number of elements automatically populated and not actual time savings. Of the total 23 metadata standard elements outlined in Table 1 above, 20 were automatically generated using the prototype method. Abstract fields typically demand some degree of intellectual forethought as to their contents; no direct attempt was therefore made to address this element. Other metadata items such as Keywords could nevertheless be used to seed the Abstract entry and guide the manual composition of said description. Relation and Source elements were similarly omitted as the test data was neither derived nor necessarily related to other existing resources; such elements could however be derived from other feature classes within the same feature dataset (Relation) or from the collection of lineage information if specifically supported (Source).

With regards a dataset's physical location and its impact on element extraction, this will be bound to the value of n, or the number of contributing levels in the adopted folder hierarchy. Increasing values of n will increase metadata element contribution but potentially complicate dataset classification and storage; the n value adopted should offer a compromise between maximising element contribution and the need to retain navigability of the storage hierarchy. A value of two was deemed appropriate for the current investigation—arguments for exceeding this in any future implementation should be carefully considered.

Feature datasets, employed to enable topological operations and impose a further degree of data organisation, must contain feature classes falling within the same geographic region and were therefore considered ideal for binding with elements depicting spatial coverage. While no similar match presented itself for feature classes, the use of the Keyword element was deemed a reasonable fit in the present context. Extending the involvement of hierarchical geodatabase components to metadata generation beyond that made by feature datasets and feature classes was not pursued. Nevertheless, internal aggregations of features supported by the geodatabase's object-relational model could conceivably be leveraged to contribute, and is consequently noted.

Initial harvesting of inherent metadata and the extraction of data properties meanwhile does demand extra effort and diligence when it comes to data preparation, but offer the additional benefit of enhancing data quality. Whilst the quality of inherent metadata by and large depends upon appropriate dataset initialisation (registering the correct projection details for example), data property extraction relies on the completion of dataset variables which may or not be required within an organisation's application domain. Any decision to cater for such variables will depend on whether it is intended to use metadata to market data resources and whether the adopted metadata standard supports an equivalent element, all the while bearing in mind that an explicit declaration of 'no value' for a metadata element eliminates the uncertainty blank entries present.

Despite the success of querying and indexing techniques employed on unstructured attribute tables, it is suggested that element extraction is better facilitated if such tables conform to a standard data schema whose constituents can be consistently and reliably referenced. Here, extra implementation costs can be partially offset through the incorporation of standard-compliant data dictionaries within surveying equipment, however querying and indexing routines may be preferred in instances where spatial data standards are deemed too unwieldy to implement. A further alternative would be the introduction of a more lightweight feature-level metadata standard on which to base extraction techniques. Advances in representing attribute semantics may well hold particular relevance here, such as the contribution of ontology-based metadata as presented by Schuurman and Leszczynski (2006).

5. Discussion

An approach for enabling the rapid production of geospatial metadata has been proposed, one which demonstrated the potential opportunities geospatial data, their applications and management practices present when undertaking automated metadata generation. While the desire is to automatically populate the maximum number of elements per metadata standard used, the role of human mediators for the purposes of quality evaluation should not be overlooked. As the approach focuses on more than pure metadata record completion, that is, it is also predicated on good data preparation and management practices, there are more potential points of entry for errors, with the subsequent need for extra diligence during data creation.

It could be argued that what is presented here is not so much the automatic generation of metadata but the transfer of effort from metadata production to data management. While this is certainly, but not exclusively, the case, it is proposed as a sound metadata management model as it encourages a well-defined data storage scheme and good data preparation; it frees up authoring resources which may now be applied to descriptive metadata and quality control—a conspicuous benefit in cases where data documenters and data authors are distinct; it safeguards metadata quality (contingent on appropriate dataset categorisation and data preparation) as elements are retrieved not entered manually; and presents a significant net saving of time despite the potentially high initial investment. Geospatial metadata has long been advocated to facilitate the management of data collections; the current approach takes this one step further, using metadata standard elements to plan data filing and in the process, contribute to metadata production. And although the test configuration may be potentially perceived as construed, it is argued that data management, by definition, should adhere to a formalised, predictable structure to best facilitate data categorisation and location.

While the present tool demonstrates a functional set of options for the automatic generation of metadata, potential for extension remains. Aside from adapting it to incorporate support for the multi-user geodatabases prevalent in corporate GIS environments, further scope exists for deriving metadata elements from both the data and the host systems. Geographic extent names can be calculated from the data's geometry by overlaying it with a reference place-name and boundary dataset bundled with the tool. Scripts which track data transformations and changes at the file level may be integrated with existing provision for monitoring data provenance to produce more detailed information on a dataset's pedigree. For organisations participating in data exchange initiatives, the tool can be extended to export an interoperable version of its data along with its metadata. By coupling this output with existing metadata serving software and an open source mapping server, an inexpensive means of visualisation can be supported for organisations wishing to market their data holdings.

6. Conclusions

It has not been the intention to laud a specific proprietary software offering, or proprietary GIS in general, merely to present what has been possible to achieve extending the basic functionality one offering provides in facilitating the production of quality geospatial metadata. Whether adopted in their entirety or in piecemeal fashion, it is contended that gains are to be had in tackling metadata generation bottlenecks and data management issues with approaches based on those outlined above.

Employing ArcGIS, the coupling of metadata production with dataset workflows was enabled, as was the exploitation of existing programming frameworks for development. While open source GIS offerings have made great strides in recent times—particularly in the realm of data sharing and visualisation across the web—the lack of a mature, widely adopted production-level desktop GIS with sophisticated metadata support argued against their use. Such a surfeit may be viewed as curious considering the weight of support behind geospatial data exchange efforts dealing with standardisation, communication protocols and software, particularly considering the pivotal role metadata plays therein.

Considering the attention paid to such data sharing initiatives, one can easily get the impression that the actual generation of quality data surrogates is taken for granted; evidence from the UK's national metadata service GIgateway would suggest that this is in fact one of the major obstacles to geospatial data exchange. Obligating and incentivising the supply of geospatial metadata will only ever work up to point; the key is to encourage a change in mentality towards the process of documenting data. Automating metadata production can help facilitate this, and supports the call to expand the scope of GI exchange efforts to include geospatial metadata generation.

The continuing convergence around ISO standards can contribute here, providing a common way to encode and manipulate metadata, thus increasing the scope for interoperability amongst emerging metadata solutions. The aim should be the development of a generic tool, readily adaptable to any application domain, yet capable of easing the burden of metadata authoring irrespective of data strategy employed. Existing provision for interoperability in proprietary solutions using OGC-compliant strategies could be extended for this purpose through the support of standard APIs to allow full access to data stores regardless of origin. GML may present an option here; a better approach however would be to access data in its native form without the need for transformation and subsequent potential for data loss.

References

Anderson, J.D., Pérez-Carballo, J., 2001. The nature of indexing: how humans and machines analyze messages and texts for retrieval: part I: research, and the nature of human indexing. Information Processing and Management: An International Journal 37 (2), 231–254.

Batcheller, J.K., Gittings, B.M., 2006. Avenues for developing the UK's National Geospatial Metadata Service. In: Proceedings of the Geographical Information Science Research UK 14th Annual Conference, University of Nottingham, Nottingham, UK, pp. 259–262.

Batcheller, J.K., Gittings, B.M., Dowers, S., 2007. The performance of vector oriented data storage structures in ESRI's ArcGIS. Transactions in GIS 11 (1), 47–65.

Craven, T., 2001. DESCRIPTION meta tags in public home and linked pages. LIBRES: Library and Information Science Research Electronic Journal 11 (2).

FGDC, 1998. FGDC-STD-001-1998. Content standard for digital geospatial metadata. Federal Geographic Data Committee, Reston, VA, USA, 90pp.

Foy, F.B., 2001. Metadata made easier? Development of improved online tool for "askGIraffe". M.Sc. Thesis, University of Edinburgh, Edinburgh, 21pp.

Göbel, S., Lutze, K., 1998. Development of meta databases for geospatial data in the WWW. In: Proceedings of the Sixth ACM International Symposium on Advances in Geographic Information Systems 1998, ACM, Washington, DC, USA, pp. 94–99.

Goodchild, M.F., Haining, R.P., 2004. GIS and spatial data analysis: converging perspectives. Papers in Regional Science 83, 363–385.

Greenberg, J., 2003. Metadata generation: process, people and tools. Bulletin of the American Society for Information Science and Technology 29 (2).

Greenberg, J., 2004. Metadata extraction and harvesting: a comparison of two automatic metadata generation applications. Journal of Internet Cataloging 6 (4), 59–82.

Greenberg, J., Spurgin, K., Crystal, A., 2006. Functionalities for automatic metadata generation applications: a survey of metadata experts' opinions. International Journal of Metadata, Semantics and Ontologies 1 (1), 3–20.

Guptill, S.G., 1999. Metadata and data catalogues. In: Longley, P., Goodchild, M.F., Maguire, D.J., Rhind, D.W. (Eds.), Geographical Information Systems. Wiley, Chichester, pp. 677–692.

ISO, 2005. BS EN ISO 19115:2005. Geographic Information—Metadata. BSi British Standards, Failand, Bristol, UK, 154pp.

ISO, 2006. BS EN ISO 19106:2006. Geographic Information—Profiles. BSi British Standards, Failand, Bristol, UK, 32pp.

Kim, T.J., 1999. Metadata for geo-spatial data sharing: a comparative analysis. The Annals of Regional Science 33, 171–181.

Leiden, K., Laughery, K.R., Keller, J., French, J., Warwick, W., Wood, S.D., 2001. A Review of Human Performance Models for the Prediction of Human Error. National Aeronautics and Space Administration, Moffett Field, CA, USA, 125pp.

Liddy, E.D., Sutton, S.A., Paik, W., Allen, E., Harwell, S., Monsour, M., Turner, A., Liddy, J., 2001. Breaking the metadata generation bottleneck: preliminary findings. In: Proceedings of the First ACM/IEEE-CS Joint Conference on Digital Libraries, Roanoke, Virginia, 464pp.

Lim, E.-P., Liu, Z., Yin, M., Goh, D.H.-L., Theng, Y.-L., Ng, W.K., 2005. On organising and accessing geospatial and georeferenced web resources using the G-Portal System. Information Processing and Management: An International Journal 41 (5), 1277–1297.

Limbach, T., Krawczyk, A., Surowiec, G., 2004. Metadata lifecycle management with GIS context. In: Proceedings of the 10th EC GI & GIS Workshop, ESDI State of the Art, Warsaw, Poland.

Longhorn, R.A., 2005. Geospatial standards, interoperability, metadata semantics and spatial data infrastructure. In: NIEeS Workshop on Activating Metadata, Cambridge, UK, 23pp.

Mathys, T., 2004. The Go-Geo! Portal metadata initiatives. In: Proceedings of the Geographical Information Science Research UK 12th Annual Conference, University of East Anglia, Norwich, UK, pp. 148–154.

Moellering, H., 1992. Opportunities for use of the spatial data transfer standard at the sate and local levels. Cartography and Geographic Information Systems 19 (5), 332–334.

Qi, L., Lingling, G., Feng, H., Yong, T., 2004. A unified metadata information management framework for digital city. In: Proceedings of IEEE's Geoscience and Remote Sensing Symposium, Anchorage, Alaska, USA, pp. 4422–4424.

Schuurman, N., Leszczynski, A., 2006. Ontology-based metadata. Transactions in GIS 10 (5), 709–726.

Stephens, R., Hochgurtel, B., 2002. Visual Basic .NET and XML. Wiley, New York, USA, 530pp.

Taylor, M., 2004. Metadata—describing geospatial data. In: Nebert, D.D. (Ed.), Developing Spatial Data Infrastructures: The SDI Cookbook Version 2.0. The Global Spatial Data Infrastructure Association, pp. 24–38.

Tsou, M.-H., 2002. An operational metadata framework for searching, indexing, and retrieving distributed geographic information services on the Internet. In: Egenhofer, M., Mark, D. (Eds.), Lecture Notes in Computer Science, vol. 2478. Springer, Berlin, pp. 313–332.

West Jr., L.A., Hess, T.J., 2002. Metadata as a knowledge management tool: supporting intelligent agent and end user access to spatial data. Decision Support Systems 32, 247–264.

Appendix C – GI Projects Registry Code

add_confirm.php

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="en">
<head>
<?php
require('includes/db_connect.php');   // database connect script.
include('includes/header.html');
?>
</head>

<body>
<div id="container">
<div id="logo">
</div>

<div id="navitabs">
<h2 class="hide">Site menu:</h2>
<a class="navitab" href="http://www.gisprojects.net">Home</a><span class="hide"> |
</span>
<a class="navitab" href="search.php">Search Initiatives</a><span class="hide"> |
</span>
<a class="navitab" href="members_area.php">Members Area</a><span class="hide"> |
</span>
<a class="navitab" href="login.php">Login</a><span class="hide"> | </span>
<a class="activenavitab" href="register.php">Register</a><span class="hide"> |
</span>
<a class="navitab" href="logout.php">Logout</a><span class="hide"> | </span>
<a class="navitab" href="#">Contact</a><span class="hide"> | </span>
</div>

<div id="desc">
<h2>Geospatial Project Registry </br>- Account Registration</h2>
<p></p>
</div>

<div id="main">
<h1>Initiative Added</h1>

<p>Thank you, your information has been added to the database.
Click <a href="add_initiative.php" title="Add_Init">here</a>
to add another</p>
<p></p>
<p></br></p>
</div>

<div id="sidebar">
</div>
<?php
include('includes/footer.html');
?>


</div>
</body>
</html>
```

add_initiative.php

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="en">
<head>
<?php
include('includes/header.html');
```

```
header("Cache-Control: public, max-age=300, must-revalidate");
$offset = 60 * 60 * 24 * 3;
 $ExpStr = "Expires: " . gmdate("D, d M Y H:i:s", time() + $offset) . " GMT";
 Header($ExpStr);
require('includes/db_connect.php');
?>
<script type="text/javascript" language="JavaScript"><!--

function activate(field) {
  field.disabled=false;
  if(document.styleSheets)field.style.visibility  = 'visible';
  field.focus(); }

function last_choice(selection) {
  return selection.selectedIndex==selection.length - 1; }

function process_choice(selection,textfield) {
  if(selection.value == "Scotland") {
    activate(textfield); }
  else {
    textfield.disabled = true;
    if(document.styleSheets)textfield.style.visibility  = 'hidden';
    textfield.value = ''; }}

function valid(menu,txt) {
  if(menu.selectedIndex == 0) {
    alert('You must make a selection from the Area of Coverage menu');
    return false;}
 }
//--></script>
</head>

<body>

<div id="container">
<div id="logo"></div>
<div id="navitabs">
<h2 class="hide">Site menu:</h2>

<a class="navitab" href="http://www.gisprojects.net/">Welcome</a><span class="hide">
| </span>
<a class="navitab" href="search.php">Search Initiatives</a><span class="hide"> |
</span>
<a class="activenavitab" href="#">Add Initiative</a><span class="hide"> | </span>
<a class="navitab" href="members_area.php">Members Area</a><span class="hide"> |
</span>
<a class="navitab" href="login.php">Login</a><span class="hide"> | </span>
<a class="navitab" href="register.php">Register</a><span class="hide"> | </span>
<a class="navitab" href="logout.php">Logout</a><span class="hide"> | </span>
<a class="navitab" href="contact.html">Contact</a><span class="hide"> | </span>
</div>

<div id="desc">
<h2>Geospatial Project Registry </br>- Add Initiative</h2>
<p></p>
</div>
<?php
function process_text(){
        // no HTML tags in username, password, etc
        // A loop would be more elegant here...
        $_POST['title'] = strip_tags($_POST['title']);
        $_POST['abstract'] = strip_tags($_POST['abstract']);
        $_POST['kwords'] = strip_tags($_POST['kwords']);
        $_POST['code'] = strip_tags($_POST['code']);
        $_POST['sp_org'] = strip_tags($_POST['sp_org']);
        $_POST['categ'] = strip_tags($_POST['categ']);
        $_POST['coverage'] = strip_tags($_POST['coverage']);
        $_POST['localauth'] = strip_tags($_POST['localauth']);
        $_POST['other'] = strip_tags($_POST['other']);
        $_POST['wsite'] = strip_tags($_POST['wsite']);
        $_POST['paddr'] = strip_tags($_POST['paddr']);
        $_POST['pcode'] = strip_tags($_POST['pcode']);
        $_POST['emailp'] = strip_tags($_POST['emailp']);
```

```
        /* the rest of the information is optional, the only thing we need to check is
if they submitted a website, and if so, check the format is ok. */
        if ($_POST['wsite'] != '' & !preg_match("/^(http|ftp):///", $_POST['wsite']))
{
                $_POST['wsite'] = 'http://'.$_POST['website'];
        }
        // now we can add them to the database.
        // encrypt password $_POST['passwd'] = md5($_POST['passwd']);
        if (!get_magic_quotes_gpc()) {
                $_POST['title'] = addslashes($_POST['title']);
                $_POST['abstract'] = addslashes($_POST['abstract']);
                $_POST['kwords'] = addslashes($_POST['kwords']);
                $_POST['code'] = addslashes($_POST['code']);
                $_POST['sp_org'] = addslashes($_POST['sp_org']);
                $_POST['categ'] = addslashes($_POST['categ']);
                $_POST['coverage'] = addslashes($_POST['coverage']);
                $_POST['localauth'] = addslashes($_POST['localauth']);
                $_POST['other'] = addslashes($_POST['other']);
                $_POST['wsite'] = addslashes($_POST['wsite']);
                $_POST['paddr'] = addslashes($_POST['paddr']);
                $_POST['pcode'] = addslashes($_POST['pcode']);
                $_POST['emailp'] = addslashes($_POST['emailp']);
        }
} ?>
<div id="main">
<div id="form">

<?php
// include the php file to process the form to ensure mandatory fields are completed
include_once('includes/easyform.php');
//Ensure that the user is logged in
if($logged_in == 0) {
        die('You must log in to add an initiative. Click <a href="login.php">here</a>
to log in or <a href="login.php">register</a>.');
        }

//Settings necessary for the easyform.php processing file
        $errorindicator='<img src="images/alert.gif" width="20" height="20"
alt="Alert" title="Indicator for missing form element" border="0" />';
        $errorclass="error";
        $Javascript=true;
        $results=check();
        if($results[0]=='Errors:'){
?>
        <h2 class="errorhead">There has been an error:</h2>
        <p>You forgot to enter the following field(s)</p>
        <ul>

<?php
        foreach ($results as $i=>$e){
                if ($i>0){
                        echo "<li>$e</li>";
                }
        }
//Test to ensure that the user has included the correct ranges for the year values
if (isset($_POST['submit'])) { // if form has been submitted
        if(($_POST['StartYear']<1500)OR($_POST['StartYear']>2100)){
                echo('<li>Start Year: Please enter a year between 1500 and
2100</li>');
        }
        global $ey;
        $ey = $_POST['EndYear'];
          if(strtoupper($ey)<>'ONGOING'){
                if(($ey*1<1500)OR($ey*1>2100)){
                        echo('<li>Year of Completion: Please enter a year between 1500
and 2100 or "Ongoing".</li>');
                }
        }
}
?>
</ul>

<?php }else if ($results['title']!=''){
<!--    <h2 class="thankshead">Thank you for sending your data</h2> -->

<?php
```

```
process_text();

$ey = $_POST['EndYear'];
if(strtoupper($ey)=='ONGOING'){
        $ey=9999;
        }

//Add the data to projects database
$rec_created = date('m d, Y');
        $insert = "INSERT INTO projects (
                title,
                abstract,
                keywords,
                code,
                sponsor_org,
                category,
                cov_area,
                loc_auth,
                other_area,
                start_date,
                end_date,
                rec_created,
                website,
                postaddr,
                postcode,
                email,
                username,
                poc,
                phone)
                        VALUES (
                        '".$_POST['title']."',
                        '".$_POST['abstract']."',
                        '".$_POST['kwords']."',
                        '".$_POST['code']."',
                        '".$_POST['sp_org']."',
                        '".$_POST['categ']."',
                        '".$_POST['coverage']."',
                        '".$_POST['localauth']."',
                        '".$_POST['other']."',
                        '".$_POST['StartYear']."',
                        '$ey',
                        '$rec_created',
                        '".$_POST['website']."',
                        '".$_POST['postaddr']."',
                        '".$_POST['postcode']."',
                        '".$_POST['emailp']."',
                        '".$_SESSION['username']."',
                        '".$_POST['poc']."',
                        '".$_POST['phone']."')";
        $add_project = $db_object->query($insert);
        if (DB::isError($add_project)) {
                die($add_project->getMessage());
        }
        $db_object->disconnect();
header('Location: add_confirm.php');
}
?>

<hr />
<strong>Initiative Details</strong><br /><br />
<form action="<?php echo $_SERVER['PHP_SELF']; ?>" method="post" name="form1"
onsubmit=
 "return valid(this.coverage,this.localauth)">
  <input type="hidden" name="required" value="title,StartYear,EndYear" />
<table width="70%" border="0" cellpadding="3" cellspacing="0" name="tab1">
    <tbody>
      <tr>
                        <td><label
for="your_title"><strong>Title</strong>*:</label></td>
                                <td><?=add('<input type="text" name="title"
id="YourTitle" maxlength="100" size="78"/>')?><br /></td>
        </tr>
      </tr>
      <tr>
        <td valign="top"><strong>Abstract</strong>:</td>
```

```
        <!-- <td><textarea name="abstract" rows="10" cols="58" align="left"
wrap=soft><?php echo $_SESSION['abstract']; ?></textarea> -->
        <td><?=add('<textarea name="abstract" id="" cols="58" rows="10" align="left"
wrap=soft></textarea>')?><td>
        </td>
      </tr>
          <tr>
        <td><label for="categ"><strong>Initiative Category</strong>:</label></td>
        <td>
        <?=add('
        <select name="categ">
               <option value="">Select:</option>
               <option value="Academic">Academic</option>
               <option value="Commercial">Commercial</option>
               <option value="Government - Central">Government - Central</option>
               <option value="Government - Local">Government - Local</option>
               <option value="NGO">NGO</option>
               <option value="other">Other</option>
        </select>
        ')?>
        <font color="red" size="1"> (Use Keyword field below to specifiy further
categories if necessary)</font>
        </td>
      </tr>
      <tr>
        <td><label for="kwords"><strong>Keywords</strong>:</label></td>
        <td><?=add('<input name="kwords" id="" maxlength="100" size="58" type="text"
/>')?><font color="red" size="1"> (comma separated)</font></td>
      </tr>
      <tr>
        <td><label for="sp_org"><strong>Sponsoring
Organisation</strong>:</label></td>
        <td><?=add('<input name="sp_org" id="" maxlength="100" size="36"
type="text"/>')?><font color="red" size="1"> (e.g funding bodies, supporting
institutions etc.)</font></td>
      </tr>
      <tr>
       <td><label for="code"><strong>Initiative Code</strong>:</label></td>
        <td><?=add('<input name="code" id="" maxlength="50" size="32"
type="text"/>')?><font color="red" size="1"> (Organisation-specific code for
initiative, if applicable)</font></td>
      </tr>
</table>
<hr />

<strong>Geographical Area</strong><br /><br />
<table>
      <tr>
        <td><label for="coverage"><strong>Area of Coverage</strong>*:</label></td>
        <td>
        <select name="coverage"
onchange="process_choice(this,document.form1.localauth)">
               <option value=""> Select:</option>
               <option value="Scotland">Scotland</option>
               <option value="England">England</option>
               <option value="Northern Ireland">Northern Ireland</option>
               <option value="Wales">Wales</option>
               <option value="United Kingdom">United Kingdom</option>
               <option value="Republic of Ireland">Republic of Ireland</option>
                      <option value="Other">Other</option>
        </select>
        <font color="red" size="1"> (Geographical extent of initiative)</font>
        </td>
          <tr>
        <!-- <td>Local Authority:</td>-->
        <td></td>
        <td>
        <script type="text/javascript" language="JavaScript"><!--
               disa = ' disabled';
               if(last_choice(document.form1.coverage)) disa = '';
                      document.write('<select name="localauth"'+disa+'
onfocus="check_choice()"><option value="">Local Authority:</option><option>Aberdeen
City</option><option>Aberdeenshire</option><option>Angus</option><option>Argyll and
Bute</option><option>City of
Edinburgh</option><option>Clackmannanshire</option><option>Dumfries and
Galloway</option><option>Dundee City</option><option>East
```

```
Ayrshire</option><option>East Dunbartonshire</option><option>East
Lothian</option><option>East
Renfrewshire</option><option>Falkirk</option><option>Fife</option><option>Glasgow
City</option><option>Highland</option><option>Inverclyde</option><option>Midlothian</
option><option>Moray</option><option>North Ayrshire</option><option>North
Lanarkshire</option><option>Orkney</option><option>Perth and
Kinross</option><option>Renfrewshire</option><option>Scottish
Borders</option><option>Shetland</option><option>South Ayrshire</option><option>South
Lanarkshire</option><option>Stirling</option><option>West
Dunbartonshire</option><option>West Lothian</option><option>Western Isles</option>');
               if(disa && document.styleSheets)
                  document.form1.localauth.style.visibility  = 'hidden';
               //--></script>
               </td>
          </tr>
      <tr>
        <td><strong>Other Area</strong>:</br>(if required)</td>
        <td><?=add('<input name="other" id="" maxlength="100" size="20"
type="text"/>')?><font color="red" size="1"> (Other placenames as
necessary)</font></td>
        <td></td>
        <td></td>
      </tr>

    </tbody>
  </table>
 <hr />
<strong>Timespan</strong><br /><br />
<table width="90%" border="0" cellpadding="3" cellspacing="0" name="tab1">
       <tr>
               <td width="40%"><strong>Initiative Start Year</strong>*:</td>
                       <td><?=add('<input type="text" name="StartYear" id="StartYear"
maxlength="9" size="9"/>')?><font color="red" size="1"> (From 1500
onwards)</font></td>
               </tr>
               <tr>
                       <td><strong>Projected Year of Completion</strong>*:</td>
                       <td><?=add('<input type="text" name="EndYear" id="EndYear"
maxlength="9" size="9" >')?><font color="red" size="1"> (From 1500 to 2100, or
"Ongoing")</font></td>
               </tr>
  </table>
    <hr />
  <table width="70%" border="0" cellpadding="3" cellspacing="0" name="tab1">
    <tbody>
 <strong>Contacts</strong><br /><br />
      <tr>
        <td valign="top"><strong>Point of Contact</strong>:</td>
        <td><?=add('<input name="poc" id="PointOfContact" maxlength="50"
type="text"/>')?></td>
        <td><strong>Telephone</strong>:</td>
        <td><?=add('<input name="phone" maxlength="25" type="text"/>')?></td>
      </tr>
      <tr>
        <td valign="top"><strong>Postal Address</strong>:</td>
        <td><?=add('<textarea name="paddr"rows="3" align="left"
wrap=soft></textarea>')?></td>
        <td><strong>Website</strong>:</td>
        <td><?=add('<input name="wsite" maxlength="100" type="text"/>')?></td>
      </tr>
      <tr>
        <td><strong>Postcode</strong>:</td>
        <td><?=add('<input name="pcode" maxlength="10" type="text"/>')?></td>
        <td><strong>E-mail</strong>:</td>
        <td><?=add('<input name="emailp" maxlength="50" type="text"/>')?></td>
      </tr>
    </tbody>
  </table>
<br />
<input name="submit" value="Add Initiative" type="submit"/>
</form>
</div>
<p></p>
<p></p>
</br></br></br></br>
</div>
```

```
<div id="sidebar">
</div>

<?php
include('includes/footer.html');
?>
</div>
</body>
</html>
```

edit_details.php

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="en">
<head>
<?php
include('includes/header.html');
?>
</head>
<body>

<div id="container">
<div id="logo">
</div>
<div id="navitabs">
<h2 class="hide">Site menu:</h2>
<a class="navitab" href="http://www.gisprojects.net/">Home</a><span class="hide"> |
</span>
<a class="navitab" href="search.php">Search Initiatives</a><span class="hide"> |
</span>
<a class="activenavitab" href="members_area.php">Members Area</a><span class="hide">
| </span>
<a class="navitab" href="login.php">Login</a><span class="hide"> | </span>
<a class="navitab" href="register.php">Register</a><span class="hide"> | </span>
<a class="navitab" href="logout.php">Logout</a><span class="hide"> | </span>
<a class="navitab" href="#">Contact</a><span class="hide"> | </span>
</div>
<div id="desc">
<h2>Geospatial Project Registry </br>- Update Records</h2>
</div>
<div id="main">
<?php

// database connect script.
require 'includes/db_connect.php';

if($logged_in == 0) {
        die('You must <a href="login.php">log in</a> to edit your details.');
}
//get_data();

if (isset($_POST['submit'])) {
        if (!$_POST['fname'] | !$_POST['lname'] | !$_POST['postaddr'] |
!$_POST['postcode'] | !$_POST['email'] ) {
                        die('You did not fill in a required field. <a
href="edit_details.php">Go back</a>.');
        }

        $check = $db_object->query("SELECT username, password FROM users WHERE
username = '".$_SESSION['username']."'");

        if (DB::isError($check) || $check->numRows() == 0) {
                die('That username does not exist in our database. <a
href="edit_details.php">Go back</a>.');
        }
        $info = $check->fetchRow();

        // check passwords match
        $_POST['password'] = stripslashes($_POST['password']);
        $info['password'] = stripslashes($info['password']);
```

```
        $_POST['password'] = md5($_POST['password']);
        $pass = $_POST['password'];

        if ($_POST['password'] != $info['password']) {
                die('Incorrect password, please try again. <a
href="edit_details.php">Go back</a>.');
        }

        if ($_POST['newpass2']<>''){
                if ($_POST['newpass'] != $_POST['newpass2']) {
                        die('Your new passwords did not match. <a
href="edit_details.php">Go back</a>.');
                }
                        // encrypt password
                $_POST['newpass2'] = md5($_POST['newpass2']);
                $pass = $_POST['newpass2'];
        }
        // check e-mail format
        if (!preg_match("/.*@.*..*/", $_POST['email']) | preg_match("/(<|>)/",
$_POST['email'])) {
                die('Invalid e-mail address. <a href="edit_details.php">Go
back</a>.');
        }
        // no HTML tags in username, password, etc
        $pass = strip_tags($pass);
        $_POST['lname'] = strip_tags($_POST['lname']);
        $_POST['fname'] = strip_tags($_POST['fname']);
        $_POST['org'] = strip_tags($_POST['org']);
        $_POST['postaddr'] = strip_tags($_POST['postaddr']);
        $_POST['postcode'] = strip_tags($_POST['postcode']);
        $_POST['phone'] = strip_tags($_POST['phone']);

        /* the rest of the information is optional, the only thing we need to
        check is if they submitted a website,
        and if so, check the format is ok. */

        // now we can add them to the database.
        if (!get_magic_quotes_gpc()) {
                $pass = addslashes($pass);
                $_POST['email'] = addslashes($_POST['email']);
                $_POST['lname'] = addslashes($_POST['lname']);
                $_POST['fname'] = addslashes($_POST['fname']);
                $_POST['org'] = addslashes($_POST['org']);
                $_POST['postaddr'] = addslashes($_POST['postaddr']);
                $_POST['postcode'] = addslashes($_POST['postcode']);
        }

        /* Defines query */
        $sql_update = "UPDATE users SET ";
        $sql_update .= "fname = '" . $_REQUEST['fname'] . "', ";
        $sql_update .= "lname = '" . $_REQUEST['lname'] . "', ";
        $sql_update .= "org = '" . $_REQUEST['org'] . "', ";
        $sql_update .= "postaddr = '" . $_REQUEST['postaddr'] . "', ";
        $sql_update .= "postcode = '" . $_REQUEST['postcode'] . "', ";
        $sql_update .= "email = '" . $_REQUEST['email'] . "', ";
        $sql_update .= "phone = '" . $_REQUEST['phone'] . "', ";
        $sql_update .= "password = '" . $pass . "' ";
        $sql_update .= "WHERE (username = '" . $_SESSION['username'] . "')";

        /* Passes query to database */
        $result = mysql_query($sql_update);
        if (!$result) {
          echo("<p>Error performing update query: " . mysql_error() .
"</p>$sql_update");
          exit();
        }

        /* Prints success message */
        print "<p> Record Successfully Updated.<a href=\"members_area.php\"> Go to
Members Area</a></p>";

//      $_SESSION['username'] = $_POST['uname'];
        $_SESSION['password'] = $pass;
//      $logged_in = 1;
//      header('Location: members_area.php');
}
```

```
else{
//get_data();
//}

//function get_data() {
        /* Defines query */
        $user = $_SESSION['username'];
        $sql = "SELECT * FROM users WHERE username = '$user' ";
        /* Passes query to database */

        $result = mysql_query($sql);
        if (!$result) {
          echo("<p>Error performing query: " . mysql_error() . "</p>");
          exit();
        }

        /* creates our row array with an if statement to report errors */
        if ($row = @mysql_fetch_array($result, MYSQL_ASSOC)) {
        ?>

        <form action="<?php $_SERVER[PHP_SELF]; ?>" method="post">
        <hr />
        <strong>Registration Details</strong></br></br>

        <table width="70%" border="0" cellpadding="3" cellspacing="0" align="center">
        <!--    <tr>
                        <td width="150"><strong>Username</strong>:</td>
                        <td width="350"><input type="text" name="username" size="75"
value="<?php echo $row[username]; ?>"></td>
                </tr> -->
                <tr>
                        <td width="30%"><strong>Firstname</strong>:</td>
                        <td><input type="text" name="fname" size="30" value="<?php echo
$row[fname]; ?>"></td>
                </tr>
                <tr>
                        <td><strong>Lastname</strong>:</td>
                        <td><input type="text" name="lname" size="30" value="<?php echo
$row[lname]; ?>"></td>
                </tr>
                <tr>
                        <td><strong>Organisation</strong>:</td>
                        <td><input type="text" name="org" size="30" value="<?php echo
$row[org]; ?>"></td>
                </tr>
                <tr>
                        <td width="150" valign="top"><strong>Postal
Address</strong>:</td>
                        <td width="350"><textarea name="postaddr" rows="3" ><?php echo
$row[postaddr]; ?></textarea></td>
                </tr>
                <tr>
                        <td width="150"><strong>Post Code</strong>:</td>
                        <td width="350"><input type="text" name="postcode" value="<?php
echo $row[postcode]; ?>"></td>
                </tr>
                <tr>
                        <td width="150"><strong>Email</strong>:</td>
                        <td width="350"><input type="text" name="email" value="<?php
echo $row[email]; ?>"></td>
                </tr>
                <tr>
                        <td width="150"><strong>Telephone</strong>:</td>
                        <td width="350"><input type="text" name="phone" value="<?php
echo $row[phone]; ?>"></td>
                </tr>
                <tr>
                        <td width="150"><strong>Old Password</strong>:</td>
                        <td width="350"><input type="password" name="password"></td>
                </tr>
                <tr>
                        <td width="150"><strong>New Password</strong>:</td>
                        <td width="350"><input type="password" name="newpass"></td>
                </tr>
```

```
                <tr>
                        <td width="150"><strong>Confirm Password</strong>:</td>
                        <td width="350"><input type="password" name="newpass2"></td>
                </tr>
                <tr>
                        <td></td>
                        <td><div style="text-align: right;"><input type="submit"
name="submit" value="Update"></div></td>
                </tr>
        </table>
        </form>
        <hr />
        <div style="text-align: right;"> <a href="members_area.php">Cancel
Edit</a></div>
        <?php
        }
}
?>
</br></br>
</div>
<div id="sidebar">
</div>
<?php
include('includes/footer.html');
?>
</div>
</body>
</html>
```

edit_record.php

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="en">
<head>
<?php
include 'includes/header.html';
require 'includes/db_connect.php';

?>
<script type="text/javascript" language="JavaScript"><!--

function activate(field) {
  field.disabled=false;
  if(document.styleSheets)field.style.visibility  = 'visible';
  field.focus(); }

function last_choice(selection) {
  return selection.selectedIndex==selection.length - 1; }

function process_choice(selection,textfield) {
  if(selection.value == "Scotland") {
    activate(textfield); }
  else {
    textfield.disabled = true;
    if(document.styleSheets)textfield.style.visibility  = 'hidden';
    textfield.value = ''; }}

function valid(menu,txt) {
  if(menu.selectedIndex == 0) {
    alert('You must make a selection from the Area of Coverage menu');
    return false;}
 }
//--></script>
</head>
<body>

<div id="container">
<div id="logo">
</div>
<div id="navitabs">
<h2 class="hide">Site menu:</h2>
```

```
<a class="navitab" href="http://www.gisprojects.net/">Home</a><span class="hide"> |
</span>
<a class="navitab" href="search.php">Search Initiatives</a><span class="hide"> |
</span>
<a class="activenavitab" href="members_area.php">Members Area</a><span class="hide">
| </span>
<a class="navitab" href="login.php">Login</a><span class="hide"> | </span>
<a class="navitab" href="register.php">Register</a><span class="hide"> | </span>
<a class="navitab" href="logout.php">Logout</a><span class="hide"> | </span>
<a class="navitab" href="#">Contact</a><span class="hide"> | </span>
</div>
<div id="desc">
<h2>Geospatial Project Registry </br>- Update Records</h2>
</div>
<div id="main">
<?php
if (!$_REQUEST['Submit']) {
     html_form();
} elseif ($_REQUEST['Submit'] == "Display") {
     select_edit();
} elseif ($_REQUEST['Submit'] == "Edit") {
     get_data();
} elseif ($_REQUEST['Submit'] == "Update") {
     update_cd();
}

function html_form() {
       $sql = "SELECT * FROM projects WHERE username = '".$_SESSION['username']."';";
       //$sql = "SELECT * FROM projects";

       $result = mysql_query($sql);
       if (!$result) {
           echo("<p>Error performing query: " . mysql_error() . "</p>");
       exit();
       }
?>
</br></br>
<p>Select a Record to Review</p>
<form name="cds" method="post" action="<? echo $_SERVER['PHP_SELF']; ?>">
<select name="projid">
<?
while ($row = mysql_fetch_array($result, MYSQL_ASSOC)) {
       echo("<option value=\"" . $row["proj_id"] . "\">" . $row["proj_id"] .": ".
$row["title"] .  "</option>\n");
}
?>
</select>
<input type="submit" name="Submit" value="Display" />
</form>

<?php
}

function select_edit() {
echo $_REQUEST['proj_id'];
$sql = "SELECT * FROM projects";
$sql .= " WHERE (proj_id = '{$_POST['projid']}')";

/* Passes a Query to the Active Database */
$result = mysql_query($sql);
if (!$result) {
  echo("<p>Error performing query: " . mysql_error() . "</p>");
  exit();
}

/* Starts the table and creates headings */
/* Retrieves the rows from the query result set
and puts them into a HTML table row */
while ($row = mysql_fetch_array($result, MYSQL_ASSOC)) {
if($row["end_date"]==9999){
       $ey="Ongoing";
}
else {
               $ey=$row["end_date"];
}
       echo("<hr />");
```

```
        echo("<table>\n");
        echo("<tr><td><strong>Title</strong>:</td>");
    echo("<td>" . $row["title"] . "</td>");
    //echo("<tr>\n<td>" . $row["title"] . "</td>");
    echo("<tr><td><strong>Abstract</strong>:</td>");
    echo("<td>" . $row["abstract"] . "</td></tr>");
    echo("<tr><td><strong>Category</strong>:</td>");
    echo("<td>" . $row["category"] . "</td></tr>");
    echo("<tr><td><strong>Keywords</strong>:</td>");
    echo("<td>" . $row["keywords"] . "</td></tr>");
        echo("<tr><td><strong>Sponsor</strong>:</td>");
    echo("<td>" . $row["sponsor_org"] . "</td></tr>");
    echo("<tr><td><strong>Initiative Code</strong>:</td>");
    echo("<td>" . $row["code"] . "</td></tr>");
        echo("</table>");
        echo("<hr />");
        echo("<table>\n");
    echo("<tr><td><strong>Area of Coverage</strong>:</td>");
    echo("<td>" . $row["cov_area"] . "</td></tr>");
    echo("<tr><td><strong>Local Authority</strong>:</td>");
    echo("<td>" . $row["loc_auth"] . "</td></tr>");
    echo("<tr><td><strong>Other Area</strong>:</td>");
    echo("<td>" . $row["other_area"] . "</td><tr>");
        echo("</table>");
        echo("<hr />");
        echo("<table>\n");
    echo("<tr><td><strong>Initiative Start Year</strong>:</td>");
    echo("<td>" . $row["start_date"] . "</td></tr>");
    echo("<tr><td><strong>Projected Year of Completion</strong>:</td>");
    echo("<td>" . $ey . "</td></tr>");
        echo("</table>");
        echo("<hr />");
        echo("<table>\n");
    echo("<tr><td><strong>Point of Contact</strong>:</td>");
    echo("<td>" . $row["poc"] ."</td></tr>");
    echo("<tr><td><strong>Postal Address</strong>:</td>");
    echo("<td>" . $row["postaddr"] . "</td></tr>");
    echo("<tr><td><strong>Post Code</strong>:</td>");
    echo("<td>" . $row["postcode"] . "</td></tr>");

    echo("<tr><td><strong>Telephone</strong>:</td>");
    echo("<td>" . $row["phone"] . "</td></tr>");

    echo("<tr><td><strong>Website</strong>:</td>");
    echo("<td>" . $row["website"] . "</td></tr>");

    echo("<tr><td><strong>Email Address</strong>:</td>");
    echo("<td>" . $row["email"] . "</td></tr>");
    echo("</table>");
    echo("<div style=\"text-align: right;\"><a href=\"" . $_SERVER['PHP_SELF'] .
"?proj_id=" .$row['proj_id'] . "&Submit=Edit\">Edit Record</a></div>\n\n");

}

html_form();
}

function get_data() {
/* Defines query */
$sql = "SELECT * FROM projects WHERE proj_id = " . $_REQUEST['proj_id'] . ";";

/* Passes query to database */
$result = mysql_query($sql);
if (!$result) {
  echo("<p>Error performing query: " . mysql_error() . "</p>");
  exit();
}

/* creates our row array with an if statement to report errors */
if ($row = @mysql_fetch_array($result, MYSQL_ASSOC)) {
?>

<!-- prints out our HTML form '\"' -->
<form action="<?php $_SERVER[PHP_SELF]; ?>" method="post" name="form1" onsubmit=
 "return valid(this.coverage,this.localauth)">
```

```
<!--/* Prints out hidden proj_id - we dont put this in the HTML form
so that the uer cannot edit the Key value in error */ -->

<!-- <input type="text" name="proj_id" value="<?php echo $row[proj_id]; ?>"> -->
<hr />
<strong>Initiative Details</strong><br /><br />
<strong> Record ID</strong>:    <?php echo $row[proj_id]; ?>

<!-- /* prints out our HTML table and fields 'escaping' any double quotes '\"' */ -->

<table width="70%" border="0" cellpadding="3" cellspacing="0">
	<tr>
		<td width="150"><strong>Title</strong>:</td>
		<td width="350"><input type="text" name="title" size="75" value="<?php
echo $row[title]; ?>"></td>
	</tr>
	<td width="150"><input type="hidden" name="proj_id" value="<?php echo
$row[proj_id]; ?>"/></td>
	<tr>
		<td width="150" valign="top"><strong>Abstract</strong>:</td>
		<td width="350"><textarea name="abstract" rows="10" cols="58" ><?php
echo $row[abstract]; ?></textarea></td>
	</tr>
	<tr>
		<td width="150"><strong>Category</strong>:</td>
		<td>
			<select name="category">
				<option value="">Select:</option>
				<option>Academic</option>
				<option>Commercial</option>
				<option>Government - Central</option>
				<option>Government - Local</option>
				<option>NGO</option>
				<option>Other</option>
			</select>
			<font color="red" size="1"> (Use Keyword field below to
specifiy further categories if necessary)</font>
		</td>
	</tr>
	<tr>
		<td width="150"><strong>Keywords</strong>:</td>
		<td width="350"><input type="text" name="keywords" size="58"
value="<?php echo $row[keywords]; ?>"><font color="red" size="1"> (comma
separated)</font></td>
	</tr>
	<tr>
		<td width="150"><strong>Sponsoring Organisation(s)</strong>:</td>
		<td width="350"><input type="text" name="sponsor_org" size="36"
value="<?php echo $row[sponsor_org]; ?>"><font color="red" size="1"> (e.g funding
bodies, supporting institutions etc.)</font></td>
	</tr>
	<tr>
		<td width="150"><strong>Initiative Code</strong>:</td>
		<td width="350"><input type="text" name="code" size="32" value="<?php
echo $row[code]; ?>"><font color="red" size="1"> (Organisation-specific code for
initiative, if applicable)</font></td>
	</tr>
</table>
<hr/>
<strong>Geographical Area</strong><br /><br />
<table width="70%" border="0" cellpadding="3" cellspacing="0">
	<tr>
		<td width="25%"><strong>Area of Coverage</strong>:</td>
		<td width="75%">
			<select name="coverage"
onchange="process_choice(this,document.form1.localauth)">
				<option value="">Select:</option>
				<option value="Scotland">Scotland</option>
				<option>England</option>
				<option>Northern Ireland</option>
				<option>Wales</option>
				<option>United Kingdom</option>
				<option>Republic of Ireland</option>
				<option>Other</option>
			</select>
```

```
                        <font color="red" size="1"> (Geographical extent of
initiative)</font>
                </td>
        </tr>
        <tr>
                <td width="150"><strong>Local Authority</strong>:</td>
                <td>
                        <script type="text/javascript" language="JavaScript"><!--
                                disa = ' disabled';
                                if(last_choice(document.form1.coverage)) disa = '';
                                        document.write('<select name="localauth"'+disa+'
onfocus="check_choice()"><option value="">Local Authority:</option><option>Aberdeen
City</option><option>Aberdeenshire</option><option>Angus</option><option>Argyll and
Bute</option><option>City of
Edinburgh</option><option>Clackmannanshire</option><option>Dumfries and
Galloway</option><option>Dundee City</option><option>East
Ayrshire</option><option>East Dunbartonshire</option><option>East
Lothian</option><option>East
Renfrewshire</option><option>Falkirk</option><option>Fife</option><option>Glasgow
City</option><option>Highland</option><option>Inverclyde</option><option>Midlothian</
option><option>Moray</option><option>North Ayrshire</option><option>North
Lanarkshire</option><option>Orkney</option><option>Perth and
Kinross</option><option>Renfrewshire</option><option>Scottish
Borders</option><option>Shetland</option><option>South Ayrshire</option><option>South
Lanarkshire</option><option>Stirling</option><option>West
Dunbartonshire</option><option>West Lothian</option><option>Western Isles</option>');
                                if(disa && document.styleSheets)
                                   document.form1.localauth.style.visibility  =
'hidden';
                        //--></script>
                </td>
        <tr>
                <td width="150"><strong>Other Area</strong>:</td>
                <td width="350"><input type="text" name="other_area" value="<?php echo
$row[other_area]; ?>"><font color="red" size="1"> (Other placenames as
necessary)</font></td>
        </tr>
</table>
<hr/>
<strong>Timespan</strong><br /><br />
<table width="70%" border="0" cellpadding="3" cellspacing="0">
        <tr>
                <td width="50%"><strong>Initiative Start Year</strong>:</td>
                <td width="350"><input type="text" name="start_date" maxlength="4"
size="4" value="<?php echo $row[start_date]; ?>"><font color="red" size="1"> (From
1500 onwards)</font></td>
        </tr>
        <tr>
                <td width="150"><strong>Projected Year of Completion</strong>:
</br></td>
                <td width="350"><input type="text" name="end_date" maxlength="4"
size="4" value="<?php echo $row[end_date]; ?>"><font color="red" size="1"> ('9999'
for Ongoing)</font></td>
        </tr>
</table>
<hr/>
<strong>Contacts</strong><br /><br />
<table width="70%" border="0" cellpadding="3" cellspacing="0">
        <tr>
                <td width="25%" valign="top"><strong>Point of Contact</strong>:</td>
                <td width="100%"><input type="text" name="poc" value="<?php echo
$row[poc]; ?>"></td>
        </tr>
        <tr>
                <td width="150" valign="top"><strong>Postal Address</strong>:</td>
                <td width="350"><textarea name="postaddr" rows="3" ><?php echo
$row[postaddr]; ?></textarea></td>
        </tr>
        <tr>
                <td width="150"><strong>Post Code</strong>:</td>
                <td width="350"><input type="text" name="postcode" value="<?php echo
$row[postcode]; ?>"></td>
        </tr>
        <tr>
                <td width="150"><strong>Telephone</strong>:</td>
```

```
		<td width="350"><input type="text" name="phone" value="<?php echo 
$row[phone]; ?>"></td>
	</tr>
	<tr>
		<td width="150"><strong>Website</strong>:</td>
		<td width="350"><input type="text" name="website" value="<?php echo 
$row[website]; ?>"></td>
	</tr>
	<tr>
		<td width="150"><strong>Email</strong>:</td>
		<td width="350"><input type="text" name="email" value="<?php echo 
$row[email]; ?>"></td>
	</tr>

</table>
<div style="text-align: right;"><input type="submit" name="Submit" 
value="Update"></div>
</form>
<hr />
<div style="text-align: right;"> <a href="edit_record.php">Cancel Edit</a></div>
<?php
}
}
function vali_date($strdate,$year,$ongoing){
	if($strdate<>9999){
		if(($strdate<1500)OR($strdate>2100)){
			echo $year;
			echo('Please enter a year between 1500 and 2100');
			echo $ongoing;
			echo('<br /><a href="edit_record.php">Go back</a>.');
			die();
		}
	}
}

function update_cd() {
/* check they filled in what they supposed to, passwords matched, username isnt 
already taken, etc. */
	if (!$_POST['title'] | !$_POST['coverage'] | !$_POST['start_date'] | 
!$_POST['end_date']) {

		die('You did not fill in a required field. Please <a 
href="edit_record.php">go back</a> and complete all mandatory fields. ');
	}
$start = "Initiative Start Date: ";
$end = "Projected Year of Completion: ";
$ongoing = " or 9999 for Ongoing";
vali_date($_POST['start_date'],$start);
vali_date($_POST['end_date'],$end,$ongoing);

/* Defines query */
$sql_update = "UPDATE projects SET ";
$sql_update .= "title = '" . $_REQUEST['title'] . "', ";
$sql_update .= "abstract = '" . $_REQUEST['abstract'] . "', ";
$sql_update .= "keywords = '" . $_REQUEST['keywords'] . "', ";
$sql_update .= "code = '" . $_REQUEST['code'] . "', ";
$sql_update .= "sponsor_org = '" . $_REQUEST['sponsor_org'] . "', ";
$sql_update .= "category = '" . $_REQUEST['category'] . "', ";
$sql_update .= "cov_area = '" . $_REQUEST['coverage'] . "', ";
$sql_update .= "loc_auth = '" . $_REQUEST['localauth'] . "', ";
$sql_update .= "other_area = '" . $_REQUEST['other_area'] . "', ";
$sql_update .= "start_date = '" . $_REQUEST['start_date'] . "', ";
$sql_update .= "end_date = '" . $_REQUEST['end_date'] . "', ";
$sql_update .= "poc = '" . $_REQUEST['poc'] . "', ";
$sql_update .= "website = '" . $_REQUEST['website'] . "', ";
$sql_update .= "phone = '" . $_REQUEST['phone'] . "', ";
$sql_update .= "postaddr = '" . $_REQUEST['postaddr'] . "', ";
$sql_update .= "postcode = '" . $_REQUEST['postcode'] . "', ";
$sql_update .= "email = '" . $_REQUEST['email'] . "' ";
$sql_update .= "WHERE (proj_id = '" . $_REQUEST['proj_id'] . "')";

/* Passes query to database */
$result = mysql_query($sql_update);
if (!$result) {
  echo("<p>Error performing update query: " . mysql_error() . "</p>$sql_update");
  exit();
```

```
}
/* Prints succes message */
print "<p> Record Successfully Updated</p>";
echo("<a href=\"" . $_SERVER['PHP_SELF'] . "?proj_id=" .$row['proj_id'] .
"&Submit=Display\">Select Another Record</a>");

/* Calls get_data() function */
}
?>
<br /><br />
</div>
<div id="sidebar">
</div>
<?php

include('includes/footer.html');
?>
</div>
</div>
</body>
</html>
```

login.php

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="en">
<head>
<?php
include('includes/header.html');
?>
</head>

<body>
<div id="container">
<div id="logo">
</div>
<div id="navitabs">
<h2 class="hide">Site menu:</h2>
<a class="navitab" href="http://www.gisprojects.net/">Home</a><span class="hide"> |
</span>
<a class="navitab" href="search.php">Search Initiatives</a><span class="hide"> |
</span>
<a class="navitab" href="members_area.php">Members Area</a><span class="hide"> |
</span>
<a class="activenavitab" href="login.php">Login</a><span class="hide"> | </span>
<a class="navitab" href="register.php">Register</a><span class="hide"> | </span>
<a class="navitab" href="logout.php">Logout</a><span class="hide"> | </span>
<a class="navitab" href="#">Contact</a><span class="hide"> | </span>
</div>

<div id="desc">
<h2>Geospatial Project Registry </br>- Login</h2>
<p></p>
</div>
<div id="main">
<?php
// database connect script.
require 'includes/db_connect.php';
if($logged_in == 1) {
	die('You are already logged in, '.$_SESSION['username'].'.');
}

if (isset($_POST['submit'])) { // if form has been submitted
	/* check they filled in what they were supposed to and authenticate */
	if(!$_POST['uname'] | !$_POST['passwd']) {
		die('You did not fill in a required field. <a href="login.php"> Go
back</a>');
	}
	// authenticate.
	if (!get_magic_quotes_gpc()) {
		$_POST['uname'] = addslashes($_POST['uname']);
	}
```

```
        $check = $db_object->query("SELECT username, password FROM users WHERE
username = '".$_POST['uname']."'");

        if (DB::isError($check) || $check->numRows() == 0) {
                die('That username does not exist in our database. <a
href="login.php"> Go back</a> and try again.');
        }

        $info = $check->fetchRow();
        // check passwords match
        $_POST['passwd'] = stripslashes($_POST['passwd']);
        $info['password'] = stripslashes($info['password']);
        $_POST['passwd'] = md5($_POST['passwd']);
        if ($_POST['passwd'] != $info['password']) {
                die('Incorrect password, please <a href="login.php"> go back </a> and
try again.');
        }

        // verify login timestamp to stop user from login back in using the back
button. Set to 60 seconds
        $time = $_POST['timestamp'];
        if (time() - $time > 60) {
                die('Your session has timed out. Please <a href="login.php"> log back
in </a>. ');
        }

        // if we get here username and password are correct,
        //register session variables and set last login time.
        $date = date('m d, Y');
        $update_login = $db_object->query("UPDATE users SET last_login = '$date' WHERE
username = '".$_POST['uname']."'");
        $_POST['uname'] = stripslashes($_POST['uname']);
        $_SESSION['username'] = $_POST['uname'];
        $_SESSION['password'] = $_POST['passwd'];
        $db_object->disconnect();
        header('Location: members_area.php');
?>

<!-- <p>Welcome back <?php echo $_SESSION['username']; ?>, you are logged in.</p> -->
<?php
} else {        // if form hasnt been submitted
?>
<!--<h1>Login</h1>-->
<p>Please enter your username and password below or <strong>Register</strong> on the
tab above</p>
<form action="<?php echo $_SERVER['PHP_SELF']?>" method="post">
<table align="center" border="0" cellspacing="2" cellpadding="3">
<tr><td><strong>Username</strong>:</td><td>
<input type="text" name="uname" maxlength="40">
</td></tr>
<input type="hidden", name="timestamp" value="<?php echo time();?>"/>
<tr><td><strong>Password</strong>:</td><td>
<input type="password" name="passwd" maxlength="50">
</td></tr>
<tr><td colspan="2" align="right">
<input type="submit" name="submit" value="Login">
</td></tr>
</table>
</form>
<?php
}
?>
<p></p>
<h3></h3>
<p></br></p>
</br></br>
</div>
<?php
include('includes/footer.html');
?>
</div>
</body>
</html>
```

logout.php

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="en">
<head>
<?php
include('includes/header.html');
?>
</head>
<body>

<div id="container">
<div id="logo">
<!--<h1><a href="index.html">GIS Projects Index</a></h1> -->
</div>
<div id="navitabs">
<h2 class="hide">Site menu:</h2>
<a class="navitab" href="http://www.gisprojects.net">Home</a><span class="hide"> |
</span>
<a class="navitab" href="search.php">Search Initiatives</a><span class="hide"> |
</span>
<a class="navitab" href="members_area.php">Members Area</a><span class="hide"> |
</span>
<a class="navitab" href="login.php">Login</a><span class="hide"> | </span>
<a class="navitab" href="register.php">Register</a><span class="hide"> | </span>
<a class="activenavitab" href="logout.php">Logout</a><span class="hide"> | </span>
<a class="navitab" href="#">Contact</a><span class="hide"> | </span>
</div>
<div id="desc">
<h2>Geospatial Project Registry </br>- Logout</h2>
</div>
<div id="main">
<?php
// database connect script.
require 'includes/db_connect.php';

if ($logged_in == 0) {
die('You are not logged in so you cannot log out.');
        }
unset($_SESSION['username']);
unset($_SESSION['password']);
// kill session variables
$_SESSION = array(); // reset session array
session_destroy();   // destroy session.
echo "You have successfully logged out";
// redirect them to anywhere you like.
// If no headers are sent, send one
if (!headers_sent()) {
   header('Location: http://www.gisprojects.net');
   exit;
}
// An example using the optional file and line parameters, as of PHP 4.3.0
// Note that $filename and $linenum are passed in for later use.
// Do not assign them values beforehand.
if (!headers_sent($filename, $linenum)) {
   header('Location: http://www.gisprojects.net');
   exit;
// You would most likely trigger an error here.
} else {
   echo "\nHeaders already sent in $filename on line $linenum\n" .
         "Cannot redirect, for now please click this <a " .
         "href=\"http://www.gisprojects.net\">link</a> instead\n";
   exit;
}
?>
</div>
<?php
include('includes/footer.html');
?>
</div>
</body>
</html>
```

members_area.php

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="en">
<head>
<?php
include('includes/header.html');
?>
</head>
<body>
<div id="container">
<div id="logo">
</div>
<div id="navitabs">
<h2 class="hide">Site menu:</h2>
<a class="navitab" href="http://www.gisprojects.net">Home</a><span class="hide"> |
</span>
<a class="navitab" href="search.php">Search Initiatives</a><span class="hide"> |
</span>
<a class="activenavitab" href="members_area.php">Members Area</a><span class="hide">
| </span>
<a class="navitab" href="login.php">Login</a><span class="hide"> | </span>
<a class="navitab" href="register.php">Register</a><span class="hide"> | </span>
<a class="navitab" href="logout.php">Logout</a><span class="hide"> | </span>
<a class="navitab" href="contact.html">Contact</a><span class="hide"> | </span>
</div>
<div id="desc">
<h2>Geospatial Project Registry </br>- Members Area</h2>
</div>
<div id="main">

<?php
// database connect script.
require 'includes/db_connect.php';
if ($logged_in == 0) {
die('You must be logged in to enter the Members Area. Please <a href="login.php"> log
in </a>or <a href="register.php"> register</a>.');
       }
?>
<p>Welcome <?php echo $_SESSION['username']; ?>, you are logged in.</p>
<p>The web site includes information on projects and initiatives, rather than
datasets (datasets should be recorded at <a
ref="http://www.gigateway.org.uk">gigateway</a>).  We are principally interested in
information on organisations (private and public sector) and projects using GI in
Scotland.; our aim is to promote sharing of ideas and information, avoid duplication
and increase the benefits to all.  Please include as many or as few projects as you
wish;  if you are unsure, please include your project. Remember, the users of this
site are interested in how you have used GI, solved a problem involving GI or
developed a new system based on GI methods.  Thus it is important you record your
project from this perspective and include contact details of an appropriate person.
For example, in the case of an Alzheimers support charity, we would be interested in
a project they are running which uses GI methods to track the demand for service. We
are not interested in the person and phone number to contact if you suffer from
Alzheimers, or their chief executive. All records are subject to editing and deletion
if regarded as inappropriate for this site. </p>

<p>Please select one of the following options: </p>
<form action="<?php echo $_SERVER['PHP_SELF']; ?>" method="post">
<input type=radio name="member_option" value="own">Browse Own Records<BR>
<!--<input type=radio name="member_option" value="all">Browse All Records<BR>
<input type=radio name="member_option" value="search">Search Initiative Database<BR>
-->
<input type=radio name="member_option" value="add">Add Initiative<BR>
<input type=radio name="member_option" value="edit">Edit Own Initiative Records<BR>
<input type=radio name="member_option" value="edit_details">Edit Own Registration
Details<BR>
<p></p>
<input type=submit value="Choose">
</form>

<?php
if ($_POST['member_option'] == "own"){
```

```
        $_SESSION['search_sql'] = ("SELECT * FROM projects WHERE username =
'".$_SESSION['username']."' LIMIT");
        $_SESSION['count_sql'] = ("SELECT COUNT(title) AS numrows FROM projects WHERE
username = '".$_SESSION['username']."'");
        header("Location: results.php");
}
if ($_POST['member_option'] == "all"){
        $_SESSION['search_sql'] = ("SELECT * FROM projects LIMIT");
        $_SESSION['count_sql'] = ("SELECT COUNT(title) AS numrows FROM projects");
        header("Location: results.php");
}
if ($_POST['member_option'] == "search"){
        header("Location: search.php");
}
if ($_POST['member_option'] == "add"){
  header("Location: add_initiative.php");
}
if ($_POST['member_option'] == "edit"){
        header("Location: edit_record.php");
}
if ($_POST['member_option'] == "edit_details"){
        header("Location: edit_details.php");
}
?>
<p></p>
</br></br></br></br></br>
</div>
<div id="sidebar">
</div>
<?php
include('includes/footer.html');
?>
</div>
</body>
</html>
```

register.php

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="en">
<head>
<?php
include('includes/header.html');
?>
</head>
<body>

<div id="container">
<div id="logo">
</div>
<div id="navitabs">
<h2 class="hide">Site menu:</h2>
<a class="navitab" href="http://www.gisprojects.net">Home</a><span class="hide"> |
</span>
<a class="navitab" href="search.php">Search Initiatives</a><span class="hide"> |
</span>
<a class="navitab" href="members_area.php">Members Area</a><span class="hide"> |
</span>
<a class="navitab" href="login.php">Login</a><span class="hide"> | </span>
<a class="activenavitab" href="register.php">Register</a><span class="hide"> |
</span>
<a class="navitab" href="logout.php">Logout</a><span class="hide"> | </span>
<a class="navitab" href="#">Contact</a><span class="hide"> | </span>
</div>

<div id="desc">
<h2>Geospatial Project Registry </br>- Account Registration</h2>
<p></p>
</div>

<div id="main">
<?php
```

```
require('includes/db_connect.php');   // database connect script.

// include the php file to process the form to ensure mandatory fields are completed
include_once('includes/easyform.php');

//Settings necessary for the easyform.php processing file
        $errorindicator='<img src="images/alert.gif" width="20" height="20"
alt="Alert" title="Indicator for missing form element" border="0" />';
        $errorclass="error";
        $Javascript=true;
        $results=check();
        if($results[0]=='Errors:'){
?>
        <h2 class="errorhead">There has been an error:</h2>
        <p>You forgot to enter the following field(s)</p>
        <ul>
<?php
        foreach ($results as $i=>$e){
                if ($i>0){
                        echo "<li>$e</li>";
                }
        }
//Test to ensure that the user has included the correct ranges for the year values
if (isset($_POST['submit'])) { // if form has been submitted
        /* check they filled in what they supposed to, passwords matched, username
        isnt already taken, etc. */

        // check passwords match
        if ($_POST['Password'] != $_POST['ConfirmPassword']) {
                echo('<li>Passwords did not match.</li>');
        }

        // check e-mail format
        if (!preg_match("/.*@.*..*/", $_POST['email']) | preg_match("/(<|>)/",
$_POST['email'])) {
                echo('<li>Invalid e-mail address.</li>');
        }
}
?>
</ul>
<?php }else if ($results['Username']!=''){
// no HTML tags in username, password, etc
        $_POST['Username'] = strip_tags($_POST['Username']);
        $_POST['Password'] = strip_tags($_POST['Password']);
        $_POST['LastName'] = strip_tags($_POST['LastName']);
        $_POST['FirstName'] = strip_tags($_POST['FirstName']);
        $_POST['org'] = strip_tags($_POST['org']);
        $_POST['PostAddress'] = strip_tags($_POST['PostAddress']);
        $_POST['PostCode'] = strip_tags($_POST['PostCode']);

        /* the rest of the information is optional, the only thing we need to
        check is if they submitted a website,
        and if so, check the format is ok. */
        // now we can add them to the database.
        // encrypt password
        $_POST['Password'] = md5($_POST['Password']);

        if (!get_magic_quotes_gpc()) {
                $_POST['Password'] = addslashes($_POST['Password']);
                $_POST['email'] = addslashes($_POST['email']);
                $_POST['LastName'] = addslashes($_POST['LastName']);
                $_POST['FirstName'] = addslashes($_POST['FirstName']);
                $_POST['org'] = addslashes($_POST['org']);
                $_POST['PostAddress'] = addslashes($_POST['PostAddress']);
                $_POST['PostCode'] = addslashes($_POST['PostCode']);
        }

        // check if username exists in database.
        if (!get_magic_quotes_gpc()) {
                $_POST['Username'] = addslashes($_POST['Username']);
        }
        $name_check = $db_object->query("SELECT username FROM users WHERE username =
'".$_POST['Username']."'");
        if (DB::isError($name_check)) {
                die($name_check->getMessage());
        }
```

```
        $name_checkk = $name_check->numRows();
        if ($name_checkk != 0) {
                die('<li>Sorry, the username: <strong>'.$_POST['Username'].'</strong>
is already taken, please pick another one.</li>');
        }
        $regdate = date('m d, Y');
        $insert = "INSERT INTO users (
                        username,
                        lname,
                        fname,
                        org,
                        phone,
                        postaddr,
                        postcode,
                        email,
                        password,
                        regdate,
                        last_login)
                        VALUES (
                        '".$_POST['Username']."',
                        '".$_POST['FirstName']."',
                        '".$_POST['LastName']."',
                        '".$_POST['org']."',
                        '".$_POST['phone']."',
                        '".$_POST['PostAddress']."',
                        '".$_POST['PostCode']."',
                        '".$_POST['email']."',
                        '".$_POST['Password']."',
                        '$regdate',
                        'Never')";
        $add_member = $db_object->query($insert);
        if (DB::isError($add_member)) {
                die($add_member->getMessage());
        }
        $db_object->disconnect();
        $_SESSION['username'] = $_POST['Username'];
        $_SESSION['password'] = $_POST['Password'];
        $logged_in = 1;
        header('Location: members_area.php');
?>
<?php
} //else {      // if form hasnt been submitted
//?>
<p> Please enter your details below. Mandatory items denoted by an asterix '*'<p>
<form action="<?php echo $_SERVER['PHP_SELF']; ?>" method="post">
        <input type="hidden" name="required"
value="FirstName,LastName,email,PostAddress,PostCode,Username,Password,ConfirmPasswor
d" />
        <table align="center" border="0" cellspacing="2" cellpadding="3">
                <tr>
                        <td><strong>First name</strong>*:</td>
                        <td><?=add('<input type="text" name="FirstName" id="FirstName"
maxlength="25">')?></td>
                </tr>
                <tr>
                        <td><strong>Last name</strong>*:</td>
                        <td><?=add('<input type="text" name="LastName" id="LastName"
maxlength="25">')?></td>
                </tr>
                <tr>
                        <td><strong>Organisation</strong>:</td>
                        <td><?=add('<input type="text" name="org"
maxlength="100">')?></td>
                </tr>
                <tr>
                        <td><strong>E-Mail</strong>*:</td>
                        <td><?=add('<input type="text" name="email" id="email"
maxlength="100">')?></td>
                </tr>
                <tr>
                        <td><strong>Contact Telephone</strong>:</td>
                        <td><?=add('<input type="text" name="phone"
maxlength="100">')?></td>
                </tr>
                <tr>
```

```
                        <td><strong>Postal Address</strong>*:</td>
                        <td><?=add('<textarea name="PostAddress" id="PostAddress"
rows="3" cols="18" align="left" wrap=soft></textarea>')?></td>
                </tr>
                <tr>
                        <td><strong>Post Code</strong>*:</td>
                        <td><?=add('<input type="text" name="PostCode" id="PostCode"
maxlength="40">')?></td>
                </tr>
                <tr>
                        <td><strong>Username</strong>*:</td>
                        <td><?=add('<input type="text" name="Username" id="Username"
maxlength="40">')?></td>
                </tr>
                <tr>
                        <td><strong>Password</strong>*:</td>
                        <td><?=add('<input type="password" name="Password"
id="Password" maxlength="50">')?></td>
                </tr>
                <tr>
                        <td><strong>Confirm Password</strong>*:</td>
                        <td><?=add('<input type="password" name="ConfirmPassword"
id="ConfirmPassword" maxlength="50">')?></td>
                </tr>
                <tr>
                        <td colspan="2" align="right"><input type="submit"
name="submit" value="Sign Up"></td>
                </tr>
        </table>
</form>
<?php
//}
?>
<p></br></p>
</div>
<div id="sidebar">
</div>

<?php
include('includes/footer.html');
?>
</div>
</body>
</html>
```

results.php

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="en">
<head>
<?php
include('includes/header.html');
?>
</head>
<body>
<div id="container">
<div id="logo">
</div>
<div id="navitabs">
<h2 class="hide">Site menu:</h2>
<a class="navitab" href="http://www.gisprojects.net">Home</a><span class="hide"> |
</span>
<a class="navitab" href="search.php">Search Initiatives</a><span class="hide"> |
</span>
<a class="activenavitab" href="#">Browse</a><span class="hide"> | </span>
<a class="navitab" href="members_area.php">Members Area</a><span class="hide"> |
</span>
<a class="navitab" href="login.php">Login</a><span class="hide"> | </span>
<a class="navitab" href="register.php">Register</a><span class="hide"> | </span>
<a class="navitab" href="logout.php">Logout</a><span class="hide"> | </span>
<a class="navitab" href="contact.html">Contact</a><span class="hide"> | </span>
</div>
```

```
<div id="desc">
<h2>Geospatial Project Registry <br>- Browse Results</h2>
<p></p>
</div>
<div id="main">
<?php
// database connect script.
require 'includes/db_connect.php';

// how many rows to show per page
$rowsPerPage = 1;

// by default we show first page
$pageNum = 1;

// if $_GET['page'] defined, use it as page number
if(isset($_GET['page']))
{
    $pageNum = $_GET['page'];
}

// counting the offset
$offset = ($pageNum - 1) * $rowsPerPage;
//$query  = "SELECT title,abstract,keywords FROM projects LIMIT $offset,
$rowsPerPage";
$query = $_SESSION['search_sql'] ." ".$offset .",".$rowsPerPage;
echo $query;

//$result = mysql_query($query) or die('Error, search_sql query1 failed ');
$result = mysql_query($query);
if (!$result) {
  die("<p>Error performing update query: " . mysql_error() . "</p>$sql_update");
}

// print the random numbers
while($row = mysql_fetch_array($result))
{
        $poc=$row['poc'];
        $id = $row['proj_id'];
        $title = $row['title'];
        $abstract = nl2br($row['abstract']);
        $keywords = $row['keywords'];
        $code = $row['code'];
        $sponsor_org = $row['sponsor_org'];
        $category = $row['category'];
        $cov_area = $row['cov_area'];
        $loc_auth = $row['loc_auth'];
        $other = $row['other_area'];
        $sdate = $row['start_date'];
        $phone = $row['phone'];
                if($row["end_date"]==9999){
                        $edate="Ongoing";
                }
                else {
                        $edate=$row["end_date"];
                }
        $username=$row['proj_un'];
        $website = $row['website'];
        $postaddr = $row['postaddr'];
        $postcode = $row['postcode'];
        $email = $row['email'];
        $project_details .=<<<EOD
        <hr/>
        <br />
        <strong>Initiative Details</strong>
        <table width="95%" border="0" cellpadding="3" cellspacing="0" align="center">
        <tr>
                <td width="15%"><strong>Title:</strong></td>
                <td>$title</td>
        </tr>
        <br />
        <tr>
                <td><strong>Record ID:</strong></td>
                <td>$id$uname</td>
```

```
        </tr>
        <br />
        <tr>
                <td><strong>Abstract:</strong></td
        <br />
                <td>$abstract</td>
        <tr>
                <td><strong>Initiative Category: </strong></td>
                <td>$category</td>
        </tr>
        <tr>
                <td><strong>Keywords: </strong></td>
                <td>$keywords</td>
        </tr>
        <tr>
                <td><strong>Sponsoring Organisation: </strong></td>
                <td>$sponsor_org</td>
        </tr>
        <tr>
                <td><strong>Initiative Code:</strong></td>
                <td>$code</td>
        </tr>
</table>
<br />
<hr/>
<br />
<strong>Geographical Area</strong>
<p></p>
<table width="95%" border="0" cellpadding="3" cellspacing="0" align="center">
        <tr>
                <td width="25%"><strong>Area of Coverage: </strong></td>
                <td>$cov_area</td>
        </tr>
        <tr>
                <td><strong>Local Authority: <br>(Scotland): </strong></td>
                <td>$loc_auth</td>
        </tr>
        <tr>
                <td><strong>Other Area of Coverage: </strong></td>
                <td>$other</td>
        </tr>
</table>
<br />
<hr/>
<br />
<strong>Timespan</strong>
<p></p>
<table width="95%" border="0" cellpadding="3" cellspacing="0" align="center">
        <tr>
                <td width="25%"><strong>Initiative Start Year: </strong></td>
                <td>$sdate</td>
        </tr>
        <tr>
                <td><strong>Projected End Year: </strong></td>
                <td>$edate</td>
        </tr>
</table>
<br />
<hr/>
<br />
<strong>Initiative Contacts</strong>
<p></p>
<table width="95%" border="0" cellpadding="3" cellspacing="0" align="center">
        <tr>
                <td width="20%"><strong>Point of Contact</strong>:</td>
                <td>$poc</td>
        </tr>
        <tr>
                <td width="20%"><strong>Postal Address</strong>:</td>
                <td>$postaddr</td>
        </tr>
        <tr>
                <td><strong>Post Code</strong>:</td>
                <td>$postcode</td>
        </tr>
        <tr>
```

```
                <td><strong>Telephone</strong>:</td>
                <td>$phone</td>
        </tr>
        <tr>
                <td><strong>Website</strong>:</td>
                <td>$website</td>
        </tr>
        <tr>
                <td><strong>Email</strong>:</td>
                <td>$email</td>
        </tr>

EOD;
$project_footer ="</table><p></p><hr/>";
$project =<<<PROJECT
                        $project_header
                        $project_details
                        $project_footer
PROJECT;
print $project;
}
if (isset($username)){
$custodian = "SELECT * from USERS where users.username='$username'";
echo "<br />";
$result = mysql_query($custodian) or die('failed');
while($row1 = mysql_fetch_array($result)){
        $ufname=$row1['fname'];
        $ulname=$row1['lname'];
        $org=$row1['org'];
        $uemail=$row1['email'];
        $uphone=$row1['phone'];
        echo('<br /><strong>Record Custodian</strong><br />');
        echo('<table width="95%" border="0" cellpadding="3" cellspacing="0"
align="center">');
        echo('<tr><td width="20%"><strong>Name:</strong></td><td>');
        echo $ufname." ".$ulname;
        echo('</td><tr width="10%"><td ><strong>Organisation:</strong></td><td>');
        echo $org;
        echo('</td><tr width="10%"><td ><strong>Telephone:</strong></td><td>');
        echo $uphone;
        echo('</td><tr width="10%"><td ><strong>Email:</strong></td><td>');
        echo $uemail;
        echo '</td><br /></table><p></p><hr />';
        }
}
// how many rows we have in database
$query = $_SESSION['count_sql'];
$result = mysql_query($query) or die('Error, query failed');

$row = mysql_fetch_array($result, MYSQL_ASSOC);

// pop the last value of the array into the numrows variable.
// Used due to inconsistent results w/ freetext searches.
// This should work as long as there is only one value in the array.
$numrows = array_pop($row);

// how many pages we have when using paging?
$maxPage = ceil($numrows/$rowsPerPage);
$self = $_SERVER['PHP_SELF'];

// creating 'previous' and 'next' link
// plus 'first page' and 'last page' link
echo '<center>';
// print 'previous' link only if we're not
// on page one
if ($pageNum > 1)
{
    $page = $pageNum - 1;
    $prev = " <a href=\"$self?page=$page\">[Prev]</a> ";
    $first = " <a href=\"$self?page=1\">[First Page]</a> ";
}
else
{
    $prev  = ' [Prev] ';       // we're on page one, don't enable 'previous' link
    $first = ' [First Page] '; // nor 'first page' link
}
```

```
// print 'next' link only if we're not
// on the last page
if ($pageNum < $maxPage)
{
    $page = $pageNum + 1;
    $next = " <a href=\"$self?page=$page\">[Next]</a> ";

    $last = " <a href=\"$self?page=$maxPage\">[Last Page]</a> ";
}
else
{
    $next = ' [Next] ';      // we're on the last page, don't enable 'next' link
    $last = ' [Last Page] '; // nor 'last page' link
}

// print the page navigation link
echo $first . $prev . " Showing page <strong>$pageNum</strong> of
<strong>$maxPage</strong> pages " . $next . $last;

echo '</center>';
?>

<p></p>
<h3></h3>
<p></br></p>
</br></br>
</div>
<div id="sidebar">
</div>
<?php
include('includes/footer.html');
?>
</div>
</body>
</html>
```

search.php

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="en">
<head>
<?php
include('includes/header.html');
?>
</head>
<body>
<div id="container">
<div id="logo">
</div>
<div id="navitabs">
<h2 class="hide">Site menu:</h2>
<a class="navitab" href="http://www.gisprojects.net/">Home</a><span class="hide"> |
</span>
<a class="activenavitab" href="search.php">Search Initiatives</a><span class="hide">
| </span>
<a class="navitab" href="members_area.php">Members Area</a><span class="hide"> |
</span>
<a class="navitab" href="login.php">Login</a><span class="hide"> | </span>
<a class="navitab" href="register.php">Register</a><span class="hide"> | </span>
<a class="navitab" href="logout.php">Logout</a><span class="hide"> | </span>
<a class="navitab" href="contact.html">Contact</a><span class="hide"> | </span>
</div>
<div id="desc">
<h2>Geospatial Project Registry </br>- Search Initiatives</h2>
</div>
<div id="main">

<?php
// Full-Text Search Example
// Conect to the database.
require('includes/db_connect.php');
```

```
// Create the search function for the freetext search:
function freeSearchForm()
{
  // Re-usable form
  // Forced boolean search. Uncomment relevant lines below and in switch statement if
norma and boolean required

  // variable setup for the form.
  $searchwords = (isset($_GET['words']) ?
htmlspecialchars(stripslashes($_REQUEST['words'])) : '');
        $boolean = 'boolean';

  echo '<form method="get" action="'.$_SERVER['PHP_SELF'].'">';
  echo '<input type="hidden" name="cmd" value="search" />';
  echo 'Search for: <input type="text" name="words" value="'.$searchwords.'" /> ';
echo '<input type="submit" value="Search" />';
  echo '</form>';
}

// The search function for specific fields
function searchByField()
{
  $searchWord = (isset($_GET['searchTerm']) ?
htmlspecialchars(stripslashes($_REQUEST['searchTerm'])) : '');
  $searchkeywords = (isset($_GET['keys']) ?
htmlspecialchars(stripslashes($_REQUEST['keys'])) : '');

  $proj_id = (($_GET['field'] == 'proj_id') ? ' selected="selected"' : '' );
  $title = (($_GET['field'] == 'title') ? ' selected="selected"' : '' );
  $keywords = (($_GET['field'] == 'keywords') ? ' selected="selected"' : '' );
  $code = (($_GET['field'] == 'code') ? ' selected="selected"' : '' );
  $sponsor_org = (($_GET['field'] == 'sponsor_org') ? ' selected="selected"' : '' );
  $category = (($_GET['field'] == 'category') ? ' selected="selected"' : '' );
  $cov_area = (($_GET['field'] == 'cov_area') ? ' selected="selected"' : '' );
  $loc_auth = (($_GET['field'] == 'loc_auth') ? ' selected="selected"' : '' );
  $other_area = (($_GET['field'] == 'other_area') ? ' selected="selected"' : '' );
  $start_date = (($_GET['field'] == 'start_date') ? ' selected="selected"' : '' );
 $end_date = (($_GET['field'] == 'end_date') ? ' selected="selected"' : '' );
  $username = (($_GET['field'] == 'username') ? ' selected="selected"' : '' );
  $lname = (($_GET['field'] == 'lname') ? ' selected="selected"' : '' );
  $postaddr = (($_GET['field'] == 'postaddr') ? ' selected="selected"' : '' );
  $postcode = (($_GET['field'] == 'postcode') ? ' selected="selected"' : '' );
        echo '<form method="get" action="'.$_SERVER['PHP_SELF'].'">';
        echo '<input type="hidden" name="cmd2" value="searchField" />';
        //echo 'Select a Field: ';
        echo '<select name="field">';
        echo '<option value="*">Select a Field:</option>';
        echo '<option value="proj_id"'.$proj_id.'>Record ID</option>';
        echo '<option value="title"'.$title.'>Title</option>';
        echo '<option value="keywords"'.$keywords.'>Keywords</option>';
        echo '<option value="code"'.$code.'>Project Code</option>';
        echo '<option value="sponsor_org"'.$sponsor_org.'>Sponsoring
Organisation</option>';
        echo '<option value="category"'.$category.'>Initiative Category</option>';
        echo '<option value="cov_area"'.$cov_area.'>Area of Coverage</option>';
        echo '<option value="loc_auth"'.$loc_auth.'>Local Authority
(Scotland)</option>';
        echo '<option value="other_area"'.$other_area.'>Other Area</option>';
        echo '<option value="start_date"'.$start_date.'>Start year</option>';
        echo '<option value="end_date"'.$end_date.'>End year</option>';
        echo '<option value="username"'.$username.'>Username</option>';
        echo '<option value="poc"'.$lname.'>Point of Contact</option>';
        echo '<option value="postaddr"'.$postaddr.'>Postal Address</option>';
        echo '<option value="postcode"'.$postcode.'>Post Code</option>';
        echo 'Search For:';
        echo 'Search for: <input type="text" name="searchTerm" value="'.$searchWord.'"
/> ';
        echo '<input type="submit" value="Search" />';
        echo '</form>';
}

function sendQuery($sql){
        $subquery = "SELECT * FROM ($sql) as table1 LIMIT";
        $subcount = "SELECT COUNT(*) FROM (SELECT * FROM ($sql) as table2) as table3";
    $_SESSION['search_sql'] = $subquery;
```

```
    $_SESSION['count_sql'] = $subcount;
    header("Location: results.php");
}
// Create the navigation switch
$cmd = (isset($_GET['cmd']) ? $_GET['cmd'] : '');
switch($cmd)
{
  default:
    echo '<h3>Free Text Search</h3>';
    echo 'Enter any number of search terms. Searches performed on Initiative Title,
Abstract and Keywords.';
    freeSearchForm();
  break;
  case "search":
    freeSearchForm();
    echo '<h3>Search Results:</h3><br />';
    $searchstring = mysql_escape_string($_GET['words']);

$sql = "SELECT proj_id, title, abstract, keywords, username proj_un,
               MATCH(title, abstract, keywords)
               AGAINST ('$searchstring' IN BOOLEAN MODE) AS score FROM projects
               WHERE MATCH(title, abstract, keywords)
               AGAINST ('$searchstring' IN BOOLEAN MODE)";
sendQuery($sql);
  break;
}

// Create the navigation switch
$cmd2 = (isset($_GET['cmd2']) ? $_GET['cmd2'] : '');
$field = $_GET['field'];
switch($cmd2)
{
  default:
    echo '<h3>Search by Field</h3>';
    echo 'Select the field you wish to search and a single search term.';
    searchByField();
  break;
  case "searchField":
       searchByField();
       $searchIDstring = mysql_escape_string($_GET['searchTerm']);
       $field = $_GET['field'];
       $sql = "SELECT proj_id, title, abstract, category, keywords, sponsor_org,
code, cov_area, loc_auth, other_area, start_date, end_date, poc, postaddr, postcode,
phone, website, email, username proj_un FROM projects WHERE $field LIKE
'%$searchIDstring%'";
       sendQuery($sql);
       break;
}
// Create the navigation switch
$cmd3 = (isset($_GET['cmd3']) ? $_GET['cmd3'] : '');
$searchyear = $_GET['range'];
$operator = $_GET['op'];
$year = (isset($_GET['year']) ? htmlspecialchars(stripslashes($_REQUEST['year'])) :
'');
       if(strtoupper($year)=='ONGOING'){
               $year = 9999;
               }
switch($cmd3)
{
       default:
               echo '<h3>Search by Start / End Year</h3>';
               echo 'Search on the basis of an initiative\'s start year or projected
year of completion.<br /><br />';
               echo '<form method="get" action="'.$_SERVER['PHP_SELF'].'">';
               echo '<input type="hidden" name="cmd3" value="searchYear" />';
?>
               <select name="range">
               <option value="start_date">Start Year</option>
               <option value="end_date">End Year</option>
               </select>
               <select name="op">
               <option value="=">Equals</option>
               <option value=">">Greater Than</option>
               <option value="<">Less Than</option>
               <option value=">=">Greater Than or Equals</option>
               <option value="<=">Less Than or Equals</option>
```

```
               </select>
<!-- echo '<input type="text" name="year" value=""/>'; -->
               <input type="text" name="year"/>
<?php
               echo '<input type="submit" value="Search" name="timespan"/><br /><font
color="red" size="1"> (Values between 1500 and 2100. Select \'End Year Equals
Ongoing\' for all ongoing initiatives)</font>';
               echo '</form>';
               break;
       case "searchYear":
               $sql = ("SELECT proj_id, title, abstract, category, keywords,
sponsor_org, code, cov_area, loc_auth, other_area, start_date, end_date, poc,
postaddr,postcode, phone, website, email, username proj_un FROM projects where
$searchyear$operator$year");
               sendQuery($sql);
               break;
}
echo '<h3>Or Browse all Records:</h3>';
echo '<form method="POST" action="'.$_SERVER['PHP_SELF'].'">';
echo '<input type="submit" value="Browse All" name="browse"/>';
echo '</form>';
if ($_POST['browse'] == "Browse All"){
       $sql = ("SELECT proj_id, title, abstract, category, keywords, sponsor_org,
code, cov_area, loc_auth, other_area, start_date, end_date, poc, postaddr, postcode,
phone, website, email, username proj_un FROM projects");
       sendQuery($sql);
}
?>
</br></br></br>
</div>
<div id="sidebar">
</div>
<?php
include('includes/footer.html');
?>
</div>
</body>
</html>
```

Appendix D – Metadata prototype code

GeminiEditor.vb

```
Imports ESRI.ArcGIS.Catalog
Imports ESRI.ArcGIS.CatalogUI
Imports ESRI.ArcGIS.Geodatabase
Imports ESRI.ArcGIS.Utility
Imports ESRI.ArcGIS.Utility.CATIDs
Imports ESRI.ArcGIS.Framework
Imports System.Runtime.InteropServices
Imports System
Imports System.IO
Imports System.Collections
Imports System.Windows.Forms

<ComClass(GeminiEditor.ClassId, GeminiEditor.InterfaceId, GeminiEditor.EventsId)> _
Public NotInheritable Class GeminiEditor
    Inherits ESRI.ArcGIS.Utility.BaseClasses.BaseCommand
    Private m_app As IGxApplication
#Region "COM GUIDs"
    ' These  GUIDs provide the COM identity for this class
    ' and its COM interfaces. If you change them, existing
    ' clients will no longer be able to access the class.
    Public Const ClassId As String = "ECA32030-98E8-4D0D-AE35-9D4B121FE044"
    Public Const InterfaceId As String = "A01C5101-D6BE-4607-B077-2730EDC3B5AE"
    Public Const EventsId As String = "4CF087A5-7B84-4665-A1F4-F045EE798443"
#End Region

#Region "Component Category Registration"
    <ComRegisterFunction(), ComVisibleAttribute(False)> _
    Public Shared Sub RegisterFunction(ByVal regKey As String)
        GxCommands.Register(regKey)
    End Sub
    <ComUnregisterFunction(), ComVisibleAttribute(False)> _
    Public Shared Sub UnregisterFunction(ByVal regKey As String)
        GxCommands.Unregister(regKey)
    End Sub
#End Region

    ' A creatable COM class must have a Public Sub New()
    ' with no parameters, otherwise, the class will not be
    ' registered in the COM registry and cannot be created
    ' via CreateObject.
    Public Sub New()

        MyBase.New()
        MyBase.m_category = "Metadata"
        MyBase.m_caption = "Gemini Metadata Editor"
        MyBase.m_message = "Edit Metadata in Gemini Format"
        MyBase.m_toolTip = "Gemini Metadata Editor"
        MyBase.m_name = "Gemini Metadata Editor"

        Dim res() As String =
GetType(GeminiEditor).Assembly.GetManifestResourceNames()
        If (res.GetLength(0) > 0) Then
            MyBase.m_bitmap = New
System.Drawing.Bitmap(GetType(GeminiEditor).Assembly.GetManifestResourceStream(res(0)
))
        End If
    End Sub
    Public Overrides Sub OnCreate(ByVal hook As Object)
        If Not (hook Is Nothing) Then
            If TypeOf (hook) Is IGxApplication Then
                m_app = hook
            Else
                MsgBox("This command is written for use in ArcCatalog only.",
vbOKOnly, "Add Subtypes")
                Exit Sub
            End If
```

```
        End If
    End Sub

    Public Overrides Sub OnClick()

        Dim pGxObject As IGxObject
        Dim pGxDataset As IGxDataset
        Dim pFeatureClass As IFeatureClass
        Dim pSubtypes As ISubtypes
        Dim intResponse As Integer
        On Error GoTo ErrorHandler

        ' Make sure application object has been set.
        If m_app Is Nothing Then Exit Sub

        ' Get the selected object.
        pGxObject = m_app.SelectedObject

        ' Make sure there is a selection.
        If pGxObject Is Nothing Then Exit Sub

        'Make sure folder has not been selected
        If TypeOf pGxObject Is IGxFolder Then
            MsgBox("You selected a folder. Please select a valid file!",
vbExclamation, "Gemini Metadata Editor")
            Exit Sub
            'Make sure disk connection has not been selected
        ElseIf TypeOf pGxObject Is IGxDiskConnection Then
            MsgBox("You selected a disk connection. Please select a valid file!",
vbExclamation, "Gemini Metadata Editor")
            Exit Sub
            'Make sure root has not been selected
        ElseIf pGxObject.BaseName = "Catalog" Then
            MsgBox("You selected Catalog root. Please select a valid file!",
vbExclamation, "Gemini Metadata Editor")
            Exit Sub
        End If

        'Catch All- Make sure a dataset is selected.
        If Not TypeOf pGxObject Is IGxDataset Then
            MsgBox("No feature class is selected.", vbOKOnly, "Add Sutypes")
            Exit Sub
        End If

        ' Set the dataset object.
        pGxDataset = pGxObject

        ' Make sure the dataset is a single featureclass.
        If Not pGxDataset.Type = esriDatasetType.esriDTFeatureClass Then
            MsgBox("No feature class is selected.", vbOKOnly, "Add Sutypes")
            Exit Sub
        End If

        '' Set the featureclass object.
        'pFeatureClass = pGxDataset.Dataset

        'Check whether metadata can be edited
        Dim pMD As IMetadata
        Dim pMDEdit As IMetadataEdit
        pMD = pGxDataset
        pMDEdit = pMD

        If pMDEdit.CanEditMetadata = False Then
            MsgBox("The associated Metadata cannot be Edited")
            Exit Sub
        End If

        ' Show the form.
        'GeminiEditorUI.FeatureClass = pFeatureClass
        Dim frm As New GeminiEditorUI
        frm.GXObj = pGxObject
        frm.Show()
        'ReadElementsFromCSV("D:\vbnet\textFolder\metaElements.csv")

        'Show DC form
        'Dim frm2 As New dceditor
```

```
        'frm2.Show()

        Exit Sub
ErrorHandler:
        MsgBox("Error: " & Err.Number & vbCrLf & Err.Description, vbExclamation,
"clsAddSubtypes:OnClick()")
        Exit Sub
    End Sub

    Public Function ReadElementsFromCSV(ByVal csvPath As String, ByVal colNo As
Integer) As ArrayList
        Dim objReader As New StreamReader(csvPath)
        Dim sLine As String = ""
        Dim arrText As New ArrayList
        Dim token As String = ""
        Dim arrToken As New ArrayList

        Do
            sLine = objReader.ReadLine()
            If Not sLine Is Nothing Then
                arrText.Add(sLine)
            End If
        Loop Until sLine Is Nothing
        objReader.Close()

        'Call the SplitString function to extract the elements needed.
        For Each sLine In arrText
            'MsgBox(sLine)
            'colNo here denotes column number to split, starting from zero
            token = SplitString(sLine, colNo)
            arrToken.Add(token)
        Next
        Return (arrToken)
    End Function

    Function SplitString(ByVal csvString As String, ByVal index As Integer) As String

        Dim tokens As String() = Nothing
        'Dim m_elementArray As New ArrayList

        tokens = csvString.Split(",")
        Return (tokens(index))
    End Function
End Class
```

GeminiEditorUI.vb

```
Imports ESRI.ArcGIS.Geodatabase
Imports ESRI.ArcGIS.CatalogUI
Imports ESRI.ArcGIS.Catalog
Imports ESRI.ArcGIS.esriSystem
Imports ESRI.ArcGIS.Utility
Imports ESRI.ArcGIS.Geometry
Imports GeminiEditor
Imports System.Reflection
Imports System.Windows.Forms
Imports System
Imports System.IO
Imports System.Xml
Imports System.Xml.XPath
Imports System.Xml.Schema
Imports System.Collections
Imports System.Drawing.Image
Imports System.Windows.Forms.AxHost
Imports System.Management
Imports System.Globalization
Imports System.Runtime.InteropServices

Public Class GeminiEditorUI
    'Inherits System.Windows.Forms.Form
    Inherits System.Windows.Forms.Form

#Region " Windows Form Designer generated code "
```

```
Public Sub New()
    MyBase.New()

    'This call is required by the Windows Form Designer.
    InitializeComponent()
    'Add any initialization after the InitializeComponent() call
End Sub

'Form overrides dispose to clean up the component list.
Protected Overloads Overrides Sub Dispose(ByVal disposing As Boolean)
    If disposing Then
        If Not (components Is Nothing) Then
            components.Dispose()
        End If
    End If
    MyBase.Dispose(disposing)
End Sub

'Required by the Windows Form Designer
Private components As System.ComponentModel.IContainer
'NOTE: The following procedure is required by the Windows Form Designer
'It can be modified using the Windows Form Designer.
'Do not modify it using the code editor.
Friend WithEvents TextBox1 As System.Windows.Forms.TextBox
Friend WithEvents TextBox5 As System.Windows.Forms.TextBox
Friend WithEvents TextBox8 As System.Windows.Forms.TextBox
Friend WithEvents TextBox11 As System.Windows.Forms.TextBox
Friend WithEvents TextBox12 As System.Windows.Forms.TextBox
Friend WithEvents TextBox13 As System.Windows.Forms.TextBox
Friend WithEvents TextBox14 As System.Windows.Forms.TextBox
Friend WithEvents TextBox15 As System.Windows.Forms.TextBox
Friend WithEvents TextBox17 As System.Windows.Forms.TextBox
Friend WithEvents TextBox18 As System.Windows.Forms.TextBox
Friend WithEvents TextBox19 As System.Windows.Forms.TextBox
Friend WithEvents TextBox20 As System.Windows.Forms.TextBox
Friend WithEvents TextBox21 As System.Windows.Forms.TextBox
Friend WithEvents TextBox22 As System.Windows.Forms.TextBox
Friend WithEvents Label1 As System.Windows.Forms.Label
Friend WithEvents Label2 As System.Windows.Forms.Label
Friend WithEvents Label4 As System.Windows.Forms.Label
Friend WithEvents Label5 As System.Windows.Forms.Label
Friend WithEvents Label6 As System.Windows.Forms.Label
Friend WithEvents Label7 As System.Windows.Forms.Label
Friend WithEvents Label8 As System.Windows.Forms.Label
Friend WithEvents Label10 As System.Windows.Forms.Label
Friend WithEvents Label11 As System.Windows.Forms.Label
Friend WithEvents Label12 As System.Windows.Forms.Label
Friend WithEvents Label13 As System.Windows.Forms.Label
Friend WithEvents Label14 As System.Windows.Forms.Label
Friend WithEvents Label15 As System.Windows.Forms.Label
Friend WithEvents Label16 As System.Windows.Forms.Label
Friend WithEvents Label17 As System.Windows.Forms.Label
Friend WithEvents Label20 As System.Windows.Forms.Label
Friend WithEvents Label21 As System.Windows.Forms.Label
Friend WithEvents Label32 As System.Windows.Forms.Label
Friend WithEvents Label33 As System.Windows.Forms.Label
Friend WithEvents Label39 As System.Windows.Forms.Label
Friend WithEvents TextBox2307 As System.Windows.Forms.TextBox
Friend WithEvents Label46 As System.Windows.Forms.Label
Friend WithEvents btnGEMINI As System.Windows.Forms.Button
Friend WithEvents TextBox701 As System.Windows.Forms.TextBox
Friend WithEvents Label47 As System.Windows.Forms.Label
Friend WithEvents TextBox702 As System.Windows.Forms.TextBox
Friend WithEvents Label3 As System.Windows.Forms.Label
Friend WithEvents GroupBox1 As System.Windows.Forms.GroupBox
Friend WithEvents RichTextBox1 As System.Windows.Forms.RichTextBox
Friend WithEvents RichTextBox2 As System.Windows.Forms.RichTextBox
Friend WithEvents RichTextBox3 As System.Windows.Forms.RichTextBox
Friend WithEvents RichTextBox4 As System.Windows.Forms.RichTextBox
Friend WithEvents MainMenu1 As System.Windows.Forms.MainMenu
Friend WithEvents MenuItem1 As System.Windows.Forms.MenuItem
Friend WithEvents MenuItem2 As System.Windows.Forms.MenuItem
Friend WithEvents MenuItem3 As System.Windows.Forms.MenuItem
Friend WithEvents MenuItem4 As System.Windows.Forms.MenuItem
Friend WithEvents MenuItem5 As System.Windows.Forms.MenuItem
```

```
    Friend WithEvents MenuItem6 As System.Windows.Forms.MenuItem
    Friend WithEvents MenuItem7 As System.Windows.Forms.MenuItem
    Friend WithEvents MenuItem8 As System.Windows.Forms.MenuItem
    Friend WithEvents MenuItem9 As System.Windows.Forms.MenuItem
    Friend WithEvents TextBox4000 As System.Windows.Forms.TextBox
    Friend WithEvents TextBox3000 As System.Windows.Forms.TextBox
    <System.Diagnostics.DebuggerStepThrough()> Private Sub InitializeComponent()
        Me.TextBox1 = New System.Windows.Forms.TextBox
        Me.TextBox5 = New System.Windows.Forms.TextBox
        Me.TextBox701 = New System.Windows.Forms.TextBox
        Me.TextBox8 = New System.Windows.Forms.TextBox
        Me.TextBox11 = New System.Windows.Forms.TextBox
        Me.TextBox12 = New System.Windows.Forms.TextBox
        Me.TextBox13 = New System.Windows.Forms.TextBox
        Me.TextBox14 = New System.Windows.Forms.TextBox
        Me.TextBox15 = New System.Windows.Forms.TextBox
        Me.TextBox17 = New System.Windows.Forms.TextBox
        Me.TextBox18 = New System.Windows.Forms.TextBox
        Me.TextBox19 = New System.Windows.Forms.TextBox
        Me.TextBox20 = New System.Windows.Forms.TextBox
        Me.TextBox21 = New System.Windows.Forms.TextBox
        Me.TextBox22 = New System.Windows.Forms.TextBox
        Me.Label1 = New System.Windows.Forms.Label
        Me.Label2 = New System.Windows.Forms.Label
        Me.Label4 = New System.Windows.Forms.Label
        Me.Label5 = New System.Windows.Forms.Label
        Me.Label6 = New System.Windows.Forms.Label
        Me.Label7 = New System.Windows.Forms.Label
        Me.Label8 = New System.Windows.Forms.Label
        Me.Label10 = New System.Windows.Forms.Label
        Me.Label11 = New System.Windows.Forms.Label
        Me.Label12 = New System.Windows.Forms.Label
        Me.Label13 = New System.Windows.Forms.Label
        Me.Label14 = New System.Windows.Forms.Label
        Me.Label15 = New System.Windows.Forms.Label
        Me.Label16 = New System.Windows.Forms.Label
        Me.Label17 = New System.Windows.Forms.Label
        Me.Label20 = New System.Windows.Forms.Label
        Me.Label21 = New System.Windows.Forms.Label
        Me.Label32 = New System.Windows.Forms.Label
        Me.Label33 = New System.Windows.Forms.Label
        Me.Label39 = New System.Windows.Forms.Label
        Me.TextBox2307 = New System.Windows.Forms.TextBox
        Me.Label46 = New System.Windows.Forms.Label
        Me.btnGEMINI = New System.Windows.Forms.Button
        Me.Label47 = New System.Windows.Forms.Label
        Me.TextBox702 = New System.Windows.Forms.TextBox
        Me.Label3 = New System.Windows.Forms.Label
        Me.GroupBox1 = New System.Windows.Forms.GroupBox
        Me.RichTextBox1 = New System.Windows.Forms.RichTextBox
        Me.RichTextBox2 = New System.Windows.Forms.RichTextBox
        Me.RichTextBox3 = New System.Windows.Forms.RichTextBox
        Me.RichTextBox4 = New System.Windows.Forms.RichTextBox
        Me.MainMenu1 = New System.Windows.Forms.MainMenu
        Me.MenuItem1 = New System.Windows.Forms.MenuItem
        Me.MenuItem2 = New System.Windows.Forms.MenuItem
        Me.MenuItem3 = New System.Windows.Forms.MenuItem
        Me.MenuItem6 = New System.Windows.Forms.MenuItem
        Me.MenuItem5 = New System.Windows.Forms.MenuItem
        Me.MenuItem4 = New System.Windows.Forms.MenuItem
        Me.MenuItem7 = New System.Windows.Forms.MenuItem
        Me.MenuItem8 = New System.Windows.Forms.MenuItem
        Me.MenuItem9 = New System.Windows.Forms.MenuItem
        Me.TextBox4000 = New System.Windows.Forms.TextBox
        Me.TextBox3000 = New System.Windows.Forms.TextBox
        Me.GroupBox1.SuspendLayout()
        Me.SuspendLayout()
        '
        'TextBox1
        '
        Me.TextBox1.Location = New System.Drawing.Point(104, 8)
        Me.TextBox1.Name = "TextBox1"
        Me.TextBox1.Size = New System.Drawing.Size(328, 20)
        Me.TextBox1.TabIndex = 2
        Me.TextBox1.Text = "health_mortality"
        '
```

```
'TextBox5
'
Me.TextBox5.Location = New System.Drawing.Point(104, 200)
Me.TextBox5.Name = "TextBox5"
Me.TextBox5.Size = New System.Drawing.Size(328, 20)
Me.TextBox5.TabIndex = 7
Me.TextBox5.Text = "Society, Health"
'
'TextBox701
'
Me.TextBox701.Location = New System.Drawing.Point(104, 232)
Me.TextBox701.Name = "TextBox701"
Me.TextBox701.Size = New System.Drawing.Size(72, 20)
Me.TextBox701.TabIndex = 8
Me.TextBox701.Text = "04 Dec 2006"
'
'TextBox8
'
Me.TextBox8.Location = New System.Drawing.Point(104, 264)
Me.TextBox8.Name = "TextBox8"
Me.TextBox8.Size = New System.Drawing.Size(72, 20)
Me.TextBox8.TabIndex = 9
Me.TextBox8.Text = "Continual"
'
'TextBox11
'
Me.TextBox11.Location = New System.Drawing.Point(104, 184)
Me.TextBox11.Name = "TextBox11"
Me.TextBox11.Size = New System.Drawing.Size(264, 20)
Me.TextBox11.TabIndex = 19
Me.TextBox11.Text = "-6.498062"
'
'TextBox12
'
Me.TextBox12.Location = New System.Drawing.Point(104, 120)
Me.TextBox12.Name = "TextBox12"
Me.TextBox12.Size = New System.Drawing.Size(264, 20)
Me.TextBox12.TabIndex = 18
Me.TextBox12.Text = "0.20327"
'
'TextBox13
'
Me.TextBox13.Location = New System.Drawing.Point(104, 88)
Me.TextBox13.Name = "TextBox13"
Me.TextBox13.Size = New System.Drawing.Size(264, 20)
Me.TextBox13.TabIndex = 17
Me.TextBox13.Text = "56.951922"
'
'TextBox14
'
Me.TextBox14.Location = New System.Drawing.Point(104, 152)
Me.TextBox14.Name = "TextBox14"
Me.TextBox14.Size = New System.Drawing.Size(264, 20)
Me.TextBox14.TabIndex = 16
Me.TextBox14.Text = "52.155362"
'
'TextBox15
'
Me.TextBox15.Location = New System.Drawing.Point(104, 24)
Me.TextBox15.Name = "TextBox15"
Me.TextBox15.Size = New System.Drawing.Size(264, 20)
Me.TextBox15.TabIndex = 15
Me.TextBox15.Text = "United Kingdom"
'
'TextBox17
'
Me.TextBox17.Location = New System.Drawing.Point(552, 72)
Me.TextBox17.Name = "TextBox17"
Me.TextBox17.Size = New System.Drawing.Size(264, 20)
Me.TextBox17.TabIndex = 13
Me.TextBox17.Text = "Restricted"
'
'TextBox18
'
Me.TextBox18.Location = New System.Drawing.Point(104, 56)
Me.TextBox18.Name = "TextBox18"
```

```
        Me.TextBox18.Size = New System.Drawing.Size(264, 20)
        Me.TextBox18.TabIndex = 12
        Me.TextBox18.Text = "GCS_Assumed_Geographic_1"
        '
        'TextBox19
        '
        Me.TextBox19.Location = New System.Drawing.Point(552, 368)
        Me.TextBox19.Name = "TextBox19"
        Me.TextBox19.Size = New System.Drawing.Size(264, 20)
        Me.TextBox19.TabIndex = 20
        Me.TextBox19.Text = ""
        '
        'TextBox20
        '
        Me.TextBox20.Location = New System.Drawing.Point(552, 40)
        Me.TextBox20.Name = "TextBox20"
        Me.TextBox20.Size = New System.Drawing.Size(264, 20)
        Me.TextBox20.TabIndex = 21
        Me.TextBox20.Text = "Vector"
        '
        'TextBox21
        '
        Me.TextBox21.Location = New System.Drawing.Point(552, 8)
        Me.TextBox21.Name = "TextBox21"
        Me.TextBox21.Size = New System.Drawing.Size(264, 20)
        Me.TextBox21.TabIndex = 22
        Me.TextBox21.Text = "Personal GeoDatabase Feature Class "
        '
        'TextBox22
        '
        Me.TextBox22.Location = New System.Drawing.Point(552, 336)
        Me.TextBox22.Name = "TextBox22"
        Me.TextBox22.Size = New System.Drawing.Size(264, 20)
        Me.TextBox22.TabIndex = 23
        Me.TextBox22.Text = "file://\\GEOSD387\D$\geospatial data root\001
continual\007 restricted\016 societ" & _
        "y.mdb"
        '
        'Label1
        '
        Me.Label1.Font = New System.Drawing.Font("Microsoft Sans Serif", 6.75!,
System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, CType(0, Byte))
        Me.Label1.ForeColor = System.Drawing.Color.Black
        Me.Label1.Location = New System.Drawing.Point(8, 8)
        Me.Label1.Name = "Label1"
        Me.Label1.Size = New System.Drawing.Size(64, 16)
        Me.Label1.TabIndex = 34
        Me.Label1.Text = "Dataset Title"
        '
        'Label2
        '
        Me.Label2.Font = New System.Drawing.Font("Microsoft Sans Serif", 6.75!,
System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, CType(0, Byte))
        Me.Label2.Location = New System.Drawing.Point(8, 40)
        Me.Label2.Name = "Label2"
        Me.Label2.Size = New System.Drawing.Size(80, 24)
        Me.Label2.TabIndex = 35
        Me.Label2.Text = "Alternative Title"
        '
        'Label4
        '
        Me.Label4.Font = New System.Drawing.Font("Microsoft Sans Serif", 6.75!,
System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, CType(0, Byte))
        Me.Label4.ForeColor = System.Drawing.Color.Black
        Me.Label4.Location = New System.Drawing.Point(8, 72)
        Me.Label4.Name = "Label4"
        Me.Label4.Size = New System.Drawing.Size(80, 24)
        Me.Label4.TabIndex = 37
        Me.Label4.Text = "Abstract"
        '
        'Label5
        '
        Me.Label5.Font = New System.Drawing.Font("Microsoft Sans Serif", 6.75!,
System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, CType(0, Byte))
        Me.Label5.ForeColor = System.Drawing.Color.Black
        Me.Label5.Location = New System.Drawing.Point(8, 200)
```

```
        Me.Label5.Name = "Label5"
        Me.Label5.Size = New System.Drawing.Size(80, 24)
        Me.Label5.TabIndex = 38
        Me.Label5.Text = " Keywords"
        '
        'Label6
        '
        Me.Label6.Font = New System.Drawing.Font("Microsoft Sans Serif", 6.75!,
System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, CType(0, Byte))
        Me.Label6.Location = New System.Drawing.Point(8, 296)
        Me.Label6.Name = "Label6"
        Me.Label6.Size = New System.Drawing.Size(80, 24)
        Me.Label6.TabIndex = 42
        Me.Label6.Text = "Dataset Creator"
        '
        'Label7
        '
        Me.Label7.Font = New System.Drawing.Font("Microsoft Sans Serif", 6.75!,
System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, CType(0, Byte))
        Me.Label7.ForeColor = System.Drawing.Color.Black
        Me.Label7.Location = New System.Drawing.Point(8, 264)
        Me.Label7.Name = "Label7"
        Me.Label7.Size = New System.Drawing.Size(80, 24)
        Me.Label7.TabIndex = 41
        Me.Label7.Text = "Update Frequency"
        '
        'Label8
        '
        Me.Label8.Font = New System.Drawing.Font("Microsoft Sans Serif", 6.75!,
System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, CType(0, Byte))
        Me.Label8.ForeColor = System.Drawing.Color.Black
        Me.Label8.Location = New System.Drawing.Point(8, 232)
        Me.Label8.Name = "Label8"
        Me.Label8.Size = New System.Drawing.Size(80, 24)
        Me.Label8.TabIndex = 40
        Me.Label8.Text = "Date Created"
        '
        'Label10
        '
        Me.Label10.Font = New System.Drawing.Font("Microsoft Sans Serif", 6.75!,
System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, CType(0, Byte))
        Me.Label10.ForeColor = System.Drawing.Color.Black
        Me.Label10.Location = New System.Drawing.Point(16, 88)
        Me.Label10.Name = "Label10"
        Me.Label10.Size = New System.Drawing.Size(80, 24)
        Me.Label10.TabIndex = 46
        Me.Label10.Text = "Northern Limit"
        '
        'Label11
        '
        Me.Label11.Font = New System.Drawing.Font("Microsoft Sans Serif", 6.75!,
System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, CType(0, Byte))
        Me.Label11.ForeColor = System.Drawing.Color.Black
        Me.Label11.Location = New System.Drawing.Point(16, 120)
        Me.Label11.Name = "Label11"
        Me.Label11.Size = New System.Drawing.Size(80, 24)
        Me.Label11.TabIndex = 45
        Me.Label11.Text = "Eastern Limit"
        '
        'Label12
        '
        Me.Label12.Font = New System.Drawing.Font("Microsoft Sans Serif", 6.75!,
System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, CType(0, Byte))
        Me.Label12.ForeColor = System.Drawing.Color.Black
        Me.Label12.Location = New System.Drawing.Point(16, 184)
        Me.Label12.Name = "Label12"
        Me.Label12.Size = New System.Drawing.Size(80, 24)
        Me.Label12.TabIndex = 44
        Me.Label12.Text = "Western Limit"
        '
        'Label13
        '
        Me.Label13.Font = New System.Drawing.Font("Microsoft Sans Serif", 6.75!,
System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, CType(0, Byte))
        Me.Label13.Location = New System.Drawing.Point(8, 424)
        Me.Label13.Name = "Label13"
```

```
        Me.Label13.Size = New System.Drawing.Size(88, 24)
        Me.Label13.TabIndex = 43
        Me.Label13.Text = "Dataset Contributor"
        '
        'Label14
        '
        Me.Label14.Font = New System.Drawing.Font("Microsoft Sans Serif", 6.75!,
System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, CType(0, Byte))
        Me.Label14.Location = New System.Drawing.Point(464, 40)
        Me.Label14.Name = "Label14"
        Me.Label14.Size = New System.Drawing.Size(80, 24)
        Me.Label14.TabIndex = 50
        Me.Label14.Text = "Dataset Type"
        '
        'Label15
        '
        Me.Label15.Font = New System.Drawing.Font("Microsoft Sans Serif", 6.75!,
System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, CType(0, Byte))
        Me.Label15.Location = New System.Drawing.Point(464, 368)
        Me.Label15.Name = "Label15"
        Me.Label15.Size = New System.Drawing.Size(64, 24)
        Me.Label15.TabIndex = 49
        Me.Label15.Text = "Relation"
        '
        'Label16
        '
        Me.Label16.Font = New System.Drawing.Font("Microsoft Sans Serif", 6.75!,
System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, CType(0, Byte))
        Me.Label16.Location = New System.Drawing.Point(16, 56)
        Me.Label16.Name = "Label16"
        Me.Label16.Size = New System.Drawing.Size(80, 24)
        Me.Label16.TabIndex = 48
        Me.Label16.Text = "Projection"
        '
        'Label17
        '
        Me.Label17.Font = New System.Drawing.Font("Microsoft Sans Serif", 6.75!,
System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, CType(0, Byte))
        Me.Label17.ForeColor = System.Drawing.Color.Black
        Me.Label17.Location = New System.Drawing.Point(464, 72)
        Me.Label17.Name = "Label17"
        Me.Label17.Size = New System.Drawing.Size(80, 24)
        Me.Label17.TabIndex = 47
        Me.Label17.Text = "Access Rights"
        '
        'Label20
        '
        Me.Label20.Font = New System.Drawing.Font("Microsoft Sans Serif", 6.75!,
System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, CType(0, Byte))
        Me.Label20.Location = New System.Drawing.Point(464, 336)
        Me.Label20.Name = "Label20"
        Me.Label20.Size = New System.Drawing.Size(64, 24)
        Me.Label20.TabIndex = 52
        Me.Label20.Text = "Identifier"
        '
        'Label21
        '
        Me.Label21.Font = New System.Drawing.Font("Microsoft Sans Serif", 6.75!,
System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, CType(0, Byte))
        Me.Label21.ForeColor = System.Drawing.Color.Black
        Me.Label21.Location = New System.Drawing.Point(464, 8)
        Me.Label21.Name = "Label21"
        Me.Label21.Size = New System.Drawing.Size(80, 24)
        Me.Label21.TabIndex = 51
        Me.Label21.Text = "Data Format"
        '
        'Label32
        '
        Me.Label32.Font = New System.Drawing.Font("Microsoft Sans Serif", 6.75!,
System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, CType(0, Byte))
        Me.Label32.ForeColor = System.Drawing.Color.Black
        Me.Label32.Location = New System.Drawing.Point(16, 24)
        Me.Label32.Name = "Label32"
        Me.Label32.Size = New System.Drawing.Size(80, 24)
        Me.Label32.TabIndex = 64
        Me.Label32.Text = "Extent Name"
```

```
        '
        'Label33
        '
        Me.Label33.Font = New System.Drawing.Font("Microsoft Sans Serif", 6.75!,
System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, CType(0, Byte))
        Me.Label33.ForeColor = System.Drawing.Color.Black
        Me.Label33.Location = New System.Drawing.Point(16, 152)
        Me.Label33.Name = "Label33"
        Me.Label33.Size = New System.Drawing.Size(80, 24)
        Me.Label33.TabIndex = 63
        Me.Label33.Text = "Southern Limit"
        '
        'Label39
        '
        Me.Label39.Font = New System.Drawing.Font("Microsoft Sans Serif", 6.75!,
System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, CType(0, Byte))
        Me.Label39.Location = New System.Drawing.Point(8, 360)
        Me.Label39.Name = "Label39"
        Me.Label39.Size = New System.Drawing.Size(88, 24)
        Me.Label39.TabIndex = 81
        Me.Label39.Text = "Dataset Publisher"
        '
        'TextBox2307
        '
        Me.TextBox2307.Location = New System.Drawing.Point(552, 400)
        Me.TextBox2307.Name = "TextBox2307"
        Me.TextBox2307.Size = New System.Drawing.Size(264, 20)
        Me.TextBox2307.TabIndex = 96
        Me.TextBox2307.Text = ""
        '
        'Label46
        '
        Me.Label46.Font = New System.Drawing.Font("Microsoft Sans Serif", 6.75!,
System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, CType(0, Byte))
        Me.Label46.Location = New System.Drawing.Point(464, 400)
        Me.Label46.Name = "Label46"
        Me.Label46.Size = New System.Drawing.Size(64, 24)
        Me.Label46.TabIndex = 95
        Me.Label46.Text = "Source"
        '
        'btnGEMINI
        '
        Me.btnGEMINI.Location = New System.Drawing.Point(760, 448)
        Me.btnGEMINI.Name = "btnGEMINI"
        Me.btnGEMINI.Size = New System.Drawing.Size(64, 32)
        Me.btnGEMINI.TabIndex = 97
        Me.btnGEMINI.Text = "Run All"
        '
        'Label47
        '
        Me.Label47.Font = New System.Drawing.Font("Microsoft Sans Serif", 6.75!,
System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, CType(0, Byte))
        Me.Label47.ForeColor = System.Drawing.Color.Black
        Me.Label47.Location = New System.Drawing.Point(192, 232)
        Me.Label47.Name = "Label47"
        Me.Label47.Size = New System.Drawing.Size(80, 24)
        Me.Label47.TabIndex = 99
        Me.Label47.Text = "Date Modified"
        '
        'TextBox702
        '
        Me.TextBox702.Location = New System.Drawing.Point(288, 232)
        Me.TextBox702.Name = "TextBox702"
        Me.TextBox702.Size = New System.Drawing.Size(72, 20)
        Me.TextBox702.TabIndex = 98
        Me.TextBox702.Text = "18 Dec 2006"
        '
        'Label3
        '
        Me.Label3.Font = New System.Drawing.Font("Microsoft Sans Serif", 6.75!,
System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, CType(0, Byte))
        Me.Label3.ForeColor = System.Drawing.Color.Black
        Me.Label3.Location = New System.Drawing.Point(8, 168)
        Me.Label3.Name = "Label3"
        Me.Label3.Size = New System.Drawing.Size(80, 24)
        Me.Label3.TabIndex = 36
```

```
        Me.Label3.Text = "Language"
        '
        'GroupBox1
        '
        Me.GroupBox1.Controls.Add(Me.Label33)
        Me.GroupBox1.Controls.Add(Me.TextBox11)
        Me.GroupBox1.Controls.Add(Me.TextBox12)
        Me.GroupBox1.Controls.Add(Me.TextBox13)
        Me.GroupBox1.Controls.Add(Me.TextBox14)
        Me.GroupBox1.Controls.Add(Me.Label10)
        Me.GroupBox1.Controls.Add(Me.Label11)
        Me.GroupBox1.Controls.Add(Me.Label12)
        Me.GroupBox1.Controls.Add(Me.Label16)
        Me.GroupBox1.Controls.Add(Me.Label32)
        Me.GroupBox1.Controls.Add(Me.TextBox15)
        Me.GroupBox1.Controls.Add(Me.TextBox18)
        Me.GroupBox1.Location = New System.Drawing.Point(448, 104)
        Me.GroupBox1.Name = "GroupBox1"
        Me.GroupBox1.Size = New System.Drawing.Size(384, 216)
        Me.GroupBox1.TabIndex = 100
        Me.GroupBox1.TabStop = False
        Me.GroupBox1.Text = "Spatial Coverage"
        '
        'RichTextBox1
        '
        Me.RichTextBox1.Location = New System.Drawing.Point(104, 72)
        Me.RichTextBox1.Name = "RichTextBox1"
        Me.RichTextBox1.Size = New System.Drawing.Size(328, 88)
        Me.RichTextBox1.TabIndex = 102
        Me.RichTextBox1.Text = ""
        '
        'RichTextBox2
        '
        Me.RichTextBox2.Location = New System.Drawing.Point(104, 296)
        Me.RichTextBox2.Name = "RichTextBox2"
        Me.RichTextBox2.Size = New System.Drawing.Size(256, 56)
        Me.RichTextBox2.TabIndex = 103
        Me.RichTextBox2.Text = "Northeastern Health Board" &
Microsoft.VisualBasic.ChrW(10) & "28 Rankeillor Street" &
Microsoft.VisualBasic.ChrW(10) & "Edinburgh EH8 9XP Scotland"
        '
        'RichTextBox3
        '
        Me.RichTextBox3.Location = New System.Drawing.Point(104, 360)
        Me.RichTextBox3.Name = "RichTextBox3"
        Me.RichTextBox3.Size = New System.Drawing.Size(256, 56)
        Me.RichTextBox3.TabIndex = 104
        Me.RichTextBox3.Text = "Northeastern Health Board" &
Microsoft.VisualBasic.ChrW(10) & "28 Rankeillor Street" &
Microsoft.VisualBasic.ChrW(10) & "Edinburgh EH8 9XP Scotland"
        '
        'RichTextBox4
        '
        Me.RichTextBox4.Location = New System.Drawing.Point(104, 424)
        Me.RichTextBox4.Name = "RichTextBox4"
        Me.RichTextBox4.Size = New System.Drawing.Size(256, 56)
        Me.RichTextBox4.TabIndex = 105
        Me.RichTextBox4.Text = "Coppock National Hospital Trust" &
Microsoft.VisualBasic.ChrW(10) & "Statistics Unit" & Microsoft.VisualBasic.ChrW(10) &
"14 Bowley Way" & Microsoft.VisualBasic.ChrW(10) & "Glasgow G4 0TT Scot" & _
        "land"
        '
        'MainMenu1
        '
        Me.MainMenu1.MenuItems.AddRange(New System.Windows.Forms.MenuItem()
{Me.MenuItem1, Me.MenuItem6, Me.MenuItem7, Me.MenuItem9})
        '
        'MenuItem1
        '
        Me.MenuItem1.Index = 0
        Me.MenuItem1.MenuItems.AddRange(New System.Windows.Forms.MenuItem()
{Me.MenuItem2, Me.MenuItem3})
        Me.MenuItem1.Text = "Generate..."
        '
        'MenuItem2
        '
```

```
        Me.MenuItem2.Index = 0
        Me.MenuItem2.Text = "Extract Metadata"
        '
        'MenuItem3
        '
        Me.MenuItem3.Index = 1
        Me.MenuItem3.Text = "Harvest XML"
        '
        'MenuItem6
        '
        Me.MenuItem6.Index = 1
        Me.MenuItem6.MenuItems.AddRange(New System.Windows.Forms.MenuItem()
{Me.MenuItem5, Me.MenuItem4})
        Me.MenuItem6.Text = "Output..."
        '
        'MenuItem5
        '
        Me.MenuItem5.Index = 0
        Me.MenuItem5.Text = "Export to XML..."
        '
        'MenuItem4
        '
        Me.MenuItem4.Index = 1
        Me.MenuItem4.Text = "Update ESRI Metadata"
        '
        'MenuItem7
        '
        Me.MenuItem7.Index = 2
        Me.MenuItem7.MenuItems.AddRange(New System.Windows.Forms.MenuItem()
{Me.MenuItem8})
        Me.MenuItem7.Text = "Configure..."
        '
        'MenuItem8
        '
        Me.MenuItem8.Index = 0
        Me.MenuItem8.Text = "Nominate Elements..."
        '
        'MenuItem9
        '
        Me.MenuItem9.Index = 3
        Me.MenuItem9.Text = "Exit"
        '
        'TextBox4000
        '
        Me.TextBox4000.Location = New System.Drawing.Point(104, 168)
        Me.TextBox4000.Name = "TextBox4000"
        Me.TextBox4000.Size = New System.Drawing.Size(112, 20)
        Me.TextBox4000.TabIndex = 106
        Me.TextBox4000.Text = "English"
        '
        'TextBox3000
        '
        Me.TextBox3000.Location = New System.Drawing.Point(104, 40)
        Me.TextBox3000.Name = "TextBox3000"
        Me.TextBox3000.Size = New System.Drawing.Size(328, 20)
        Me.TextBox3000.TabIndex = 107
        Me.TextBox3000.Text = "National, age-standardised mortality rates - United
Kingdom"
        '
        'GeminiEditorUI
        '
        Me.AutoScaleBaseSize = New System.Drawing.Size(5, 13)
        Me.BackColor = System.Drawing.SystemColors.ActiveBorder
        Me.ClientSize = New System.Drawing.Size(840, 489)
        Me.Controls.Add(Me.TextBox3000)
        Me.Controls.Add(Me.TextBox4000)
        Me.Controls.Add(Me.RichTextBox4)
        Me.Controls.Add(Me.RichTextBox3)
        Me.Controls.Add(Me.RichTextBox2)
        Me.Controls.Add(Me.RichTextBox1)
        Me.Controls.Add(Me.GroupBox1)
        Me.Controls.Add(Me.Label47)
        Me.Controls.Add(Me.TextBox702)
        Me.Controls.Add(Me.TextBox2307)
        Me.Controls.Add(Me.TextBox22)
        Me.Controls.Add(Me.TextBox21)
```

```
        Me.Controls.Add(Me.TextBox20)
        Me.Controls.Add(Me.TextBox19)
        Me.Controls.Add(Me.TextBox17)
        Me.Controls.Add(Me.TextBox8)
        Me.Controls.Add(Me.TextBox701)
        Me.Controls.Add(Me.TextBox5)
        Me.Controls.Add(Me.TextBox1)
        Me.Controls.Add(Me.btnGEMINI)
        Me.Controls.Add(Me.Label46)
        Me.Controls.Add(Me.Label39)
        Me.Controls.Add(Me.Label20)
        Me.Controls.Add(Me.Label21)
        Me.Controls.Add(Me.Label14)
        Me.Controls.Add(Me.Label15)
        Me.Controls.Add(Me.Label17)
        Me.Controls.Add(Me.Label13)
        Me.Controls.Add(Me.Label6)
        Me.Controls.Add(Me.Label7)
        Me.Controls.Add(Me.Label8)
        Me.Controls.Add(Me.Label5)
        Me.Controls.Add(Me.Label4)
        Me.Controls.Add(Me.Label3)
        Me.Controls.Add(Me.Label2)
        Me.Controls.Add(Me.Label1)
        Me.Menu = Me.MainMenu1
        Me.Name = "GeminiEditorUI"
        Me.Text = "Metadata Generator"
        Me.GroupBox1.ResumeLayout(False)
        Me.ResumeLayout(False)
    End Sub
#End Region
    Dim m_pGxObj As IGxObject
    Dim m_pGxDS As IGxDataset
    Dim m_pGxDB As IGxDatabase
    Dim m_pMetaData As IMetadata
    Dim m_xmlprops As IXmlPropertySet2

    'Custom metadata editor configuration variables
    Dim tagFile As String = "D:\vbnet\textFolder\config\crosswalk.csv"
    Dim geminiTemplate As String =
"D:\vbnet\textFolder\templates\gemini_template.xml"
    Dim fgdcTemplate As String = "D:\vbnet\textFolder\templates\fgdc_template.xml"
    Dim outputLocation As String = "D:\vbnet\textFolder\output\"
    Dim geminiFileTree As String = "D:\vbnet\textFolder\config\tree.csv"

Private Sub btnUpdate_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs)
        Dim vTags As Object
        Dim vValues As Object
        MsgBox("ERRORBICUITS!")
        'm_xmlprops.GetPropertiesByAttribute("idinfo/citation/citeinfo/title sync"
"true", True, vTags, vValues)
        'm_xmlprops.GetPropertiesByAttribute(
        MsgBox(vValues(0))

        m_xmlprops.SetAttribute("idinfo/citation/citeinfo/title", "TITLE", "Rob",
esriXmlSetPropertyAction.esriXSPAAddDuplicate)
    End Sub

Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)

        'Populate elements based on location of Feature class in the file tree
        Try
            ElementsFromDirTree()
        Catch
            MsgBox("Prob with file path calc on button click")
        End Try

        ' =*=* Get the Minimum and Maximum date values from the dataset attribute
table
        ' Used to populate "Date" element (#7)
        If (Me.TextBox701.Text = "") Then

            Dim minMaxDates As ArrayList
            Dim dateString As String
            Dim min, max As Date
```

```
            minMaxDates = ReturnMinMaxDates(m_pGxDS)
            min = minMaxDates(0)
            max = minMaxDates(1)
            If min = "" And max = "" Then
                dateString = ""
            Else
                dateString = min.ToString("yyyyMMdd") & "-" &
max.ToString("yyyyMMdd")
            End If

            Try
                PopTextBoxByTag(7, dateString)
            Catch
                MsgBox("No Date fields found in attribute table (on button click)")
            End Try
        End If

        'Complete Dataset Reference date (element #8) with current date if empty
        If (Me.TextBox8.Text = "") Then
            Try
                PopTextBoxByTag(8, System.DateTime.Now.ToString("yyyyMMdd"))
            Catch
                MsgBox("Error Reading current Date")
            End Try
        End If

        'Set Alternative title as dataset's alias
        'Not suitable for shapefiles as they do not support aliases
        'If (Me.TextBox2.Text = "") Then
        'Try
        PopTextBoxByTag(2, GetAlternativeTitle())
        'Catch
        '   MsgBox("Error Retrieving Alternative Title")
        '  End Try
        ' End If

        'Vertical Extent Info
        Dim vertExArray As ArrayList
        Dim dZmin As Double
        Dim dZmax As Double
        Dim zUnits As String
        Dim vEI As String

        'Try
        '    vertExArray = GetVerticalExtentInfo(m_pGxDS)
        '    If (Me.TextBox1601.Text = "") Then
        '        PopTextBoxByTag(1601, vertExArray(0))
        '    End If
        '    If (Me.TextBox1602.Text = "") Then
        '        PopTextBoxByTag(1602, vertExArray(1))
        '    End If
        '    If (Me.TextBox1603.Text = "") Then
        '        PopTextBoxByTag(1603, vertExArray(2))
        '    End If
        '    If (Me.TextBox1604.Text = "") Then
        '        PopTextBoxByTag(1604, vertExArray(4))
        '    End If
        'Catch
        '    MsgBox("No Vertical Extent Info Detected")
        'End Try

        'Spatial Resolution - the number of system units (stored by ArcGIS) per unit
of measure (defined by the coord system)
        If (Me.TextBox18.Text = "") Then
            Try
                PopTextBoxByTag(18, GetSpatialResolution)
            Catch
                MsgBox("Error Retrieving Spatial Resolution Information")
            End Try
        End If

    End Sub
    Public Sub GeminiEditorUI_Load(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles MyBase.Load
```

```
        ' =*=* Load all nodes beneath root (from ESRI's metadata object) into an xml
document in memory
        Dim xmlDoc As XmlDocument
        xmlDoc = LoadESRIXml("/")

        ' =*=* Load Metadata profile tags / elements into a list.
        'Tags are indexed / accessed in the order in which they are read from file
        Dim tagArray As ArrayList
        tagArray = ReadTagsFromFile(tagFile, 2) '2 denotes colNo (column number 3)
        'Marshal.ReleaseComObject()
        ' =*=* Instantiate a XPathNavigator with which nodes may be traversed
        Dim xmlNav As XPathNavigator = xmlDoc.CreateNavigator()
        PopInitial(tagArray, xmlNav)
        ' =*=* Show a thumbnail of the data on the form
        '=>ShowThumb()
        'ExportFGDC(3, tagFile)

        ' =*=* get username of current logged in user
        'MsgBox("User :" & CurrentUser())

        ' =*=* identify all blank textboxes
        Dim error As ArrayList

        error = FindBlankFields(Me)

        Dim arr As IEnumerator = error.GetEnumerator
        While arr.MoveNext
            'MsgBox(arr.Current)
        End While

        ' =*=* get system language
        'GetSystemLanguage()

        ' =*=* get regional settings
        'RegionalSettings()

        ' =*=* Get Vertical Extent Info
        'GetVerticalExtentInfo(m_pGxDS)
        'GetVerticalExtentInfo(m_pGxDS)

    End Sub

    'This property is set before the form is opened, meaning we have access to the
metadata we access when the GeminiEditor command is clicked
    Public Property GXObj()
        Get
            Return m_pGxObj
        End Get
        Set(ByVal Value)
            m_pGxObj = Value
        End Set
    End Property
Private Function LoadESRIXml(ByVal nodeStr As String) As XmlDocument
        'MSXML code and samples - needed due to inconsistent behaviour of ESRI's
xpath handling
        Dim myXmldoc As New XmlDocument

        m_pGxDS = m_pGxObj
        m_pMetaData = m_pGxDS
        m_xmlprops = m_pMetaData.Metadata()
        ' load the XML from dataIdInfo below into MSXML "document"
        myXmldoc.LoadXml(m_xmlprops.GetXml(nodeStr))
        Return myXmldoc
End Function

Function ExportFGDC()
        Dim outXmlDoc As New XmlDocument
        outXmlDoc.Load(fgdcTemplate)
        Dim standardCode As Integer = 3
        Dim tagArray2 As ArrayList
        tagArray2 = ReadTagsFromFile(tagFile, standardCode)
        Dim xNode As XmlNode
        Dim txtBox As String
        Dim txtBoxIndex As Integer
        Dim tokens As String() = Nothing
        Dim str As String
```

```
        '*** Element 1 - Title
        xNode = outXmlDoc.SelectSingleNode(tagArray2(1))
        xNode.InnerText = Me.TextBox1.Text
        '*** Element 2 - Alternative Title - Not present in gigateway fgdc profile
        '*** Element 3 - Dataset Language - Not present in gigateway fgdc profile
        '*** Element 4 - Abstract
        xNode = outXmlDoc.SelectSingleNode(tagArray2(4))
        'xNode.InnerText = Me.TextBox4.Text
        '*** Element 5 - Topic Category
        xNode = outXmlDoc.SelectSingleNode(tagArray2(5))
        str = Me.TextBox5.Text
        ParseAndCloneSibling(str, xNode)
        ''*** Element 6 - Subject - Not present in gigateway fgdc profile
        '*** Element 7 - Date
        'Start Data Capture Date
        xNode = outXmlDoc.SelectSingleNode(tagArray2(7))
        xNode.InnerText = Me.TextBox701.Text
        'End Data Capture Date
        xNode = outXmlDoc.SelectSingleNode(tagArray2(8))
        xNode.InnerText = Me.TextBox702.Text
        ''*** Element 8 - Dataset Reference Date - Not present in gigateway fgdc
profile
        '*** Element 9 - Originator Org Name 'Does not currently support multiple
originators
        xNode = outXmlDoc.SelectSingleNode(tagArray2(10))
        'xNode.InnerText = Me.TextBox901.Text & " " & Me.TextBox902.Text
        ''*** Element 10 - Lineage - Not present in gigateway fgdc profile
        '*** Elements 11-14 - Bounding Coordinates
        'Save textbox values (11-14)
        txtBoxIndex = 11
        Do Until txtBoxIndex > 14
            txtBox = "TextBox" & txtBoxIndex
            xNode = outXmlDoc.SelectSingleNode(tagArray2(txtBoxIndex + 2))
            xNode.InnerText = FindControl(txtBox, Me).Text
            txtBoxIndex += 1
        Loop
        '*** Element 15 - Extent

        xNode = outXmlDoc.SelectSingleNode(tagArray2(17))
        str = Me.TextBox15.Text
        ParseAndClone(str, xNode)

        ''*** Element 16 - Vertical Extent Information - Not present in gigateway
fgdc profile
        ''minVal
        ''maxVal
        ''UOM

        ''*** Element 17 - Spatial Reference System - Not present in gigateway fgdc
profile
        ''*** Element 18 - Spatial Resolution - Not present in gigateway fgdc profile
        ''*** Element 19 - Spatial Representation Type - Not present in gigateway
fgdc profile

        '*** Element 20 - Presentation Type
        xNode = outXmlDoc.SelectSingleNode(tagArray2(25))
        xNode.InnerText = Me.TextBox20.Text 'Multiple values stored as compound
element
        ''*** Element 21 - Data Format - Not present in gigateway fgdc profile
        ''*** Element 22 - Supply Media - Not present in gigateway fgdc profile
        '*** Element 23 - Distributor - Name 'Does not currently support multiple
distributors
        xNode = outXmlDoc.SelectSingleNode(tagArray2(28))
        'xNode.InnerText = Me.TextBox2301.Text

        'Distributor Contact POC
        xNode = outXmlDoc.SelectSingleNode(tagArray2(29))
        'xNode.InnerText = Me.TextBox2302.Text

        'Distributor Address
        Dim compoundAddr As String
        'compoundAddr = Me.TextBox2303.Text & " " & Me.TextBox2304.Text & " " &
Me.TextBox2305.Text & " " & Me.TextBox2306.Text & " " & Me.TextBox2307.Text
        xNode = outXmlDoc.SelectSingleNode(tagArray2(30))
        xNode.InnerText = compoundAddr
```

```
        'Distributor Phone
        xNode = outXmlDoc.SelectSingleNode(tagArray2(35))
        'xNode.InnerText = Me.TextBox2308.Text

        'Distributor FAX
        xNode = outXmlDoc.SelectSingleNode(tagArray2(36))
        'xNode.InnerText = Me.TextBox2309.Text

        'Distributor email
        xNode = outXmlDoc.SelectSingleNode(tagArray2(37))
        'xNode.InnerText = Me.TextBox2310.Text

        'Distributor web address - Not present in gigateway fgdc profile
        'xNode = outXmlDoc.SelectSingleNode(tagArray2(38))
        'xNode.InnerText = Me.TextBox2311.Text

        '*** Element 24 - Frequency of Update
        xNode = outXmlDoc.SelectSingleNode(tagArray2(39))
        'xNode.InnerText = Me.TextBox24.Text

        '*** Element 25 - Access Constraint
        xNode = outXmlDoc.SelectSingleNode(tagArray2(40))
        'xNode.InnerText = Me.TextBox25.Text

        '*** Element 26 - Use Constraint
        xNode = outXmlDoc.SelectSingleNode(tagArray2(41))
        'xNode.InnerText = Me.TextBox26.Text

        '*** Element 27 - Additional information source - Not present in gigateway
fgdc profile

        '*** Element 28 - Online Resource - Not present in gigateway fgdc profile

        '*** Element 29 - Browse graphic
        xNode = outXmlDoc.SelectSingleNode(tagArray2(44))
        'str = Me.TextBox29.Text
        ParseAndClone(str, xNode)

        '*** Elements 30 - 32 - Date of Update of Metadata, Standard Name & Version
        'Save textbox values (30-32)
        txtBoxIndex = 30
        Do Until txtBoxIndex > 32
            txtBox = "TextBox" & txtBoxIndex
            xNode = outXmlDoc.SelectSingleNode(tagArray2(txtBoxIndex + 15))
            xNode.InnerText = FindControl(txtBox, Me).Text
            txtBoxIndex += 1
        Loop

        Dim outfile As String
        outfile = outputLocation & Me.TextBox1.Text & "_fgdc.xml"
        outXmlDoc.Save(outfile)
        MsgBox("Save OK")

    End Function
    'Function ExportXml(ByVal xPathString As String, ByVal standardTemplate As
String)
    Function ExportGEMINI()
        Dim standardCode As Integer = 4

        Dim outXmlDoc As New XmlDocument
        outXmlDoc.Load(geminiTemplate)

        Dim tagArray2 As ArrayList
        tagArray2 = ReadTagsFromFile(tagFile, standardCode)

        'Define a lookup table with all namespaces used in the GEMINI template
        'Required for Xpath expressions
        Dim nsmgr As XmlNamespaceManager
        nsmgr = New XmlNamespaceManager(outXmlDoc.NameTable)

        nsmgr.AddNamespace("gemini", "http://www.gigateway.org.uk/gemini")
        'MsgBox("Namespace " & nsmgr.LookupNamespace("gemini"))
        nsmgr.AddNamespace("egemini", "http://www.gigateway.org.uk/egemini")
        nsmgr.AddNamespace("xs", "http://www.w3.org/2001/XMLSchema")
        nsmgr.AddNamespace("xsi", "http://www.w3.org/2001/XMLSchema-instance")
```

```
        nsmgr.AddNamespace("xlink", "http://www.w3.org/1999/xlink")
        nsmgr.AddNamespace("gml", "http://www.opengis.net/gml")
        nsmgr.AddNamespace("gts", "http://www.isotc211.org/schemas/2005/gts")
        nsmgr.AddNamespace("gco", "http://www.isotc211.org/schemas/2005/gco")
        nsmgr.AddNamespace("gts", "http://www.isotc211.org/schemas/2005/gts")
        nsmgr.AddNamespace("gmd", "http://www.isotc211.org/schemas/2005/gmd")
        nsmgr.AddNamespace("java", "http//xml.apache.org/xslt/java")
        nsmgr.AddNamespace("sflatlongconvert",
"uk.co.snowflakesoft.metaeditor.transform.ConvertToLatLongValue")

        Dim xNode As XmlNode
        Dim txtBox As String
        Dim txtBoxIndex As Integer
        Dim tokens As String() = Nothing
        Dim str As String
        MsgBox("OK")

        '*** Element 1 - Title
        xNode = outXmlDoc.SelectSingleNode(tagArray2(1), nsmgr)
        xNode.InnerText = Me.TextBox1.Text
        MsgBox("OK")

        '*** Element 2 - Alternative Title
        xNode = outXmlDoc.SelectSingleNode(tagArray2(2), nsmgr)
        'str = Me.TextBox2.Text
        ParseAndClone(str, xNode, nsmgr)
        MsgBox("OK")

        '*** Element 3 - Dataset Language
        xNode = outXmlDoc.SelectSingleNode(tagArray2(3), nsmgr)
        MsgBox("outxml OK")
        'str = Me.TextBox3.Text
        MsgBox("textbox ok")
        ParseAndClone(str, xNode, nsmgr)
        MsgBox("element 3 OK")

        '*** Element 4 - Abstract
        xNode = outXmlDoc.SelectSingleNode(tagArray2(4), nsmgr)
        'xNode.InnerText = Me.TextBox4.Text
        MsgBox("OK")

        '*** Element 5 - Topic Category
        xNode = outXmlDoc.SelectSingleNode(tagArray2(5), nsmgr)
        str = Me.TextBox5.Text
        ParseAndClone(str, xNode, nsmgr)
        MsgBox("OK")

        '*** Element 6 - Subject
        xNode = outXmlDoc.SelectSingleNode(tagArray2(5), nsmgr)
        'str = Me.TextBox6.Text
        ParseAndClone(str, xNode, nsmgr)

        '*** Element 7 - Date
        'Start Data Capture Date
        xNode = outXmlDoc.SelectSingleNode(tagArray2(7), nsmgr)
        xNode.InnerText = Me.TextBox701.Text
        'End Data Capture Date
        xNode = outXmlDoc.SelectSingleNode(tagArray2(8), nsmgr)
        xNode.InnerText = Me.TextBox702.Text

        '*** Element 8 - Datset Reference Date
        xNode = outXmlDoc.SelectSingleNode(tagArray2(9), nsmgr)
        xNode.InnerText = Me.TextBox8.Text

        '*** Element 9 - Originator Org Name 'Does not currently support multiple
originators
        xNode = outXmlDoc.SelectSingleNode(tagArray2(10), nsmgr)
        'xNode.InnerText = Me.TextBox901.Text & " " & Me.TextBox902.Text

        '*** Elements 10-14
        'Save textbox values (10-14)
        txtBoxIndex = 10
        Do Until txtBoxIndex > 14
            txtBox = "TextBox" & txtBoxIndex
            xNode = outXmlDoc.SelectSingleNode(tagArray2(txtBoxIndex + 2), nsmgr)
            xNode.InnerText = FindControl(txtBox, Me).Text
```

```
            txtBoxIndex += 1
        Loop

        '*** Element 15 - Extent
        xNode = outXmlDoc.SelectSingleNode(tagArray2(17), nsmgr)
        str = Me.TextBox15.Text
        ParseAndClone(str, xNode, nsmgr)

        '*** Element 16 - Vertical Extent Information - minVal
        xNode = outXmlDoc.SelectSingleNode(tagArray2(18), nsmgr)
        'xNode.InnerText = Me.TextBox1601.Text
        'maxVal
        xNode = outXmlDoc.SelectSingleNode(tagArray2(19), nsmgr)
        'xNode.InnerText = Me.TextBox1602.Text
        'UOM
        xNode = outXmlDoc.SelectSingleNode(tagArray2(20), nsmgr)
        'xNode.InnerText = Me.TextBox1603.Text
        'Vertical Datum
        xNode = outXmlDoc.SelectSingleNode(tagArray2(21), nsmgr)
        'xNode.InnerText = Me.TextBox1604.Text

        '*** Elements 17-18
        'Save textbox values (17-18)
        txtBoxIndex = 17
        Do Until txtBoxIndex > 18
            txtBox = "TextBox" & txtBoxIndex
            xNode = outXmlDoc.SelectSingleNode(tagArray2(txtBoxIndex + 5), nsmgr)
            xNode.InnerText = FindControl(txtBox, Me).Text
            txtBoxIndex += 1
        Loop

        '*** Element 19 - Spatial Representation Type
        xNode = outXmlDoc.SelectSingleNode(tagArray2(24), nsmgr)
        str = Me.TextBox19.Text
        ParseAndCloneByAttribute(str, "codeListValue", xNode, nsmgr)

        '*** Element 20 - Presentation Type
        xNode = outXmlDoc.SelectSingleNode(tagArray2(25), nsmgr)
        str = Me.TextBox20.Text
        ParseAndCloneByAttribute(str, "codeListValue", xNode, nsmgr)

        '*** Element 21 - Data Format
        xNode = outXmlDoc.SelectSingleNode(tagArray2(26), nsmgr)
        str = Me.TextBox21.Text
        ParseAndCloneByAttribute(str, "codeListValue", xNode, nsmgr)

        '*** Element 22 - Supply media
        xNode = outXmlDoc.SelectSingleNode(tagArray2(27), nsmgr)
        str = Me.TextBox22.Text
        ParseAndCloneByAttribute(str, "codeListValue", xNode, nsmgr)

        '*** Element 23 - Distributor - Name 'Does not currently support multiple
distributors
        xNode = outXmlDoc.SelectSingleNode(tagArray2(28), nsmgr)
        'xNode.InnerText = Me.TextBox2301.Text
        'Distributor Contact POC
        xNode = outXmlDoc.SelectSingleNode(tagArray2(29), nsmgr)
        'xNode.InnerText = Me.TextBox2302.Text
        'Distributor Address
        Dim compoundAddr As String
        'compoundAddr = Me.TextBox2303.Text & " " & Me.TextBox2304.Text & " " &
Me.TextBox2305.Text & " " & Me.TextBox2306.Text & " " & Me.TextBox2307.Text
        xNode = outXmlDoc.SelectSingleNode(tagArray2(30), nsmgr)
        xNode.InnerText = compoundAddr
        'Distributor Phone
        xNode = outXmlDoc.SelectSingleNode(tagArray2(35), nsmgr)
        'xNode.InnerText = Me.TextBox2308.Text
        'Distributor FAX
        xNode = outXmlDoc.SelectSingleNode(tagArray2(36), nsmgr)
        'xNode.InnerText = Me.TextBox2309.Text
        'Distributor email
        xNode = outXmlDoc.SelectSingleNode(tagArray2(37), nsmgr)
        'xNode.InnerText = Me.TextBox2310.Text
        'Distributor web address
        xNode = outXmlDoc.SelectSingleNode(tagArray2(38), nsmgr)
        'xNode.InnerText = Me.TextBox2311.Text
```

```
        '*** Element 24 - Frequency of Update
        xNode = outXmlDoc.SelectSingleNode(tagArray2(39), nsmgr)
        'xNode.InnerText = Me.TextBox24.Text

        '*** Element 25 - Access Constraint
        xNode = outXmlDoc.SelectSingleNode(tagArray2(40), nsmgr)
        'str = Me.TextBox25.Text
        ParseAndCloneByAttribute(str, "codeListValue", xNode, nsmgr)

        '*** Element 26 - Use Constraint
        xNode = outXmlDoc.SelectSingleNode(tagArray2(41), nsmgr)
        'str = Me.TextBox26.Text
        ParseAndCloneByAttribute(str, "codeListValue", xNode, nsmgr)

        '*** Element 27 - Additional information source
        xNode = outXmlDoc.SelectSingleNode(tagArray2(42), nsmgr)
        'xNode.InnerText = Me.TextBox27.Text

        '*** Element 28 - Online Resource
        xNode = outXmlDoc.SelectSingleNode(tagArray2(43), nsmgr)
        'str = Me.TextBox28.Text
        ParseAndCloneByAttribute(str, "codeListValue", xNode, nsmgr)

        '*** Element 29 - Browse graphic
        xNode = outXmlDoc.SelectSingleNode(tagArray2(44), nsmgr)
        'str = Me.TextBox29.Text
        ParseAndCloneByAttribute(str, "codeListValue", xNode, nsmgr)

        '*** Elements 30 - 32
        'Save textbox values (30-32)
        txtBoxIndex = 30
        Do Until txtBoxIndex > 32
            txtBox = "TextBox" & txtBoxIndex
            xNode = outXmlDoc.SelectSingleNode(tagArray2(txtBoxIndex + 15), nsmgr)
            xNode.InnerText = FindControl(txtBox, Me).Text
            txtBoxIndex += 1
        Loop

        'Console.WriteLine("Display the modified XML...")
        Dim outfile As String
        outfile = outputLocation & Me.TextBox1.Text & "_gemini.xml"
        outXmlDoc.Save(outfile)
        MsgBox("Save OK")
    End Function
    Function ParseAndCloneSibling(ByVal str As String, ByVal xNode As XmlNode)
        Dim tokens As String()
        tokens = str.Split(";")
        Dim int As Integer = tokens.Length() - 1

        If int >= 0 Then
            xNode.InnerText = tokens(0)
            While (int > 0)
                Dim clone As XmlNode = xNode.Clone
                Dim parent As XmlNode = xNode.ParentNode
                clone.FirstChild.InnerText = tokens(int)
                parent.InsertAfter(clone, xNode)
                int -= 1
            End While
        End If
    End Function

Overloads Function ParseAndClone(ByVal str As String, ByVal xNode As XmlNode, ByVal
nsmgr As XmlNamespaceManager)
        'parses multiple metadata element values (delimited by ";")
        'creates node clones and sets the node values to the parsed items
        Dim tokens As String()
        tokens = str.Split(";")
        Dim int As Integer = tokens.Length() - 1
        If int >= 0 Then
            xNode.InnerText = tokens(0)
            While (int > 0)
                Dim parent As XmlNode = xNode.ParentNode
                Dim clone As XmlNode = parent.Clone
                Dim grandfather As XmlNode = parent.ParentNode
```

```
                'Select the corresponding child node in the cloned node and set its
value
                Dim childClone As XmlNode = clone.SelectSingleNode(xNode.Name, nsmgr)
                childClone.InnerText = tokens(int)
                grandfather.InsertAfter(clone, parent)
                int -= 1
            End While
        End If
    End Function
Overloads Function ParseAndClone(ByVal str As String, ByVal xNode As XmlNode)
        'parses multiple metadata element values (delimited by ";")
        'creates node clones and sets the node values to the parsed items
        Dim tokens As String()
        tokens = str.Split(";")
        Dim int As Integer = tokens.Length() - 1
        If int >= 0 Then
            xNode.InnerText = tokens(0)
            While (int > 0)
                Dim parent As XmlNode = xNode.ParentNode
                Dim clone As XmlNode = parent.Clone
                Dim grandfather As XmlNode = parent.ParentNode
                'Select the corresponding child node in the cloned node and set its
value
                Dim childClone As XmlNode = clone.SelectSingleNode(xNode.Name)
                childClone.InnerText = tokens(int)
                grandfather.InsertAfter(clone, parent)
                int -= 1
            End While
        End If
    End Function
Function ParseAndCloneByAttribute(ByVal str As String, ByVal attr As String, ByVal
xNode As XmlNode, ByVal nsmgr As XmlNamespaceManager)
        'Similar to ParseAndClone but used for elements with multiple attributes
        'Creates a new node for each attribute value

        Dim tokens As String()
        tokens = str.Split(";")
        Dim int As Integer = tokens.Length() - 1
        'Dim x As XmlAttribute
        Dim xmlElement As XmlElement
        xmlElement = xNode
        If int >= 0 Then
            xmlElement.SetAttribute(attr, tokens(0))
            While (int > 0)
                Dim parent As XmlNode = xmlElement.ParentNode
                Dim clone As XmlNode = parent.Clone
                Dim grandParent As XmlNode = parent.ParentNode
                Dim insert As XmlElement = clone.SelectSingleNode(xNode.Name, nsmgr)
                insert.SetAttribute("codeListValue", tokens(int))
                grandParent.InsertAfter(clone, parent)
                int -= 1
            End While
        End If
    End Function
Function ReadTagsFromFile(ByVal filePath As String, ByVal colNo As Integer) As
ArrayList
        'set this to a variable read from a text file,
        'iterate thoughout placing the values in the appropriate text boxes
        Dim gemEd As New GeminiEditor
        Dim gemEdArray As New ArrayList
        Dim arrElement As String
        Dim arrCount As Integer = 1 'Start at one as the csv has a header line
        'Create a new editor, run the ReadElements code, populate an array element
paths used
        'to retrieve values stored by ArcCatalog
        gemEdArray = gemEd.ReadElementsFromCSV(filePath, colNo)
       gemEd = Nothing
        Return gemEdArray
    End Function
Function PopInitial(ByVal tagArray As ArrayList, ByVal xmlNav As XPathNavigator)
        Dim xpathResult As Object
        'Populate each GEMINI element in turn
        '***** Element 1: Title
        xpathResult = EvalXpath(xmlNav, tagArray(1))
        PopTextBoxByTag(1, xpathResult)
```

```
        '***** Element 2: Alternative title
        Try
            xpathResult = EvalXpath(xmlNav, tagArray(2))
            PopTextBoxByTag(2, xpathResult)
        Catch
            MsgBox("No Alt Title")
        End Try


        '***** Element 3: Dataset Language
        ' redo to narrow attributes returned
        xpathResult = EvalXpath(xmlNav, tagArray(3)) ' & "/@*") 'return all
attributes of the elements specified by tagArray(3)
        PopTextBoxByTag(3, xpathResult)

        '***** Element 4: Abstract
        xpathResult = EvalXpath(xmlNav, tagArray(4))
        PopTextBoxByTag(4, xpathResult)

        '***** Element 5: TopicCategory
        xpathResult = EvalXpath(xmlNav, tagArray(5)) ' & "/@*")
        PopTextBoxByTag(5, xpathResult)

        '***** Element 6: Subject
        xpathResult = EvalXpath(xmlNav, tagArray(6))
'"metadata/dataIdInfo/descKeys[@KeyTypCd='005']/keyword")
        PopTextBoxByTag(6, xpathResult)

        '***** Element 7: Date
        'Start Data Capture Date
        xpathResult = EvalXpath(xmlNav, tagArray(7))
'"metadata/dataIdInfo/idCitation/resRefDate[refDate][refDateType/DateTypCd[@value='00
3']]")
        PopTextBoxByTag(701, xpathResult)
        'End Data Capture Date
        xpathResult = EvalXpath(xmlNav, tagArray(8))
        PopTextBoxByTag(702, xpathResult)

        '***** Element 8: Dataset Reference Date
        xpathResult = EvalXpath(xmlNav, tagArray(9))
'"metadata/dataIdInfo/idCitation/resRefDate[refDate][refDateType/DateTypCd[@value='00
2']]")
        PopTextBoxByTag(8, xpathResult)

        '***** Element 9: Originator
        'Originator Organisation Name
        xpathResult = EvalXpath(xmlNav, tagArray(10))
'"metadata/dataIdInfo/idCitation/citRespParty[role/RoleCd[@value='010']]/child::*")
        PopTextBoxByTag(901, xpathResult)
        'Originator Organisation POC
        xpathResult = EvalXpath(xmlNav, tagArray(11))
'"metadata/dataIdInfo/idCitation/citRespParty[role/RoleCd[@value='010']]/child::*")
        PopTextBoxByTag(902, xpathResult)

        '***** Element 10: Lineage
        xpathResult = EvalXpath(xmlNav, tagArray(12))
        'xpathResult = EvalXpath(xmlNav,
"metadata/Esri/DataProperties/lineage/child::*")
        PopTextBoxByTag(10, xpathResult)

        '***** Element 11: West BL
        xpathResult = EvalXpath(xmlNav, tagArray(13))
        PopTextBoxByTag(11, xpathResult)

        '***** Element 12: East BL
        xpathResult = EvalXpath(xmlNav, tagArray(14))
        PopTextBoxByTag(12, xpathResult)

        '***** Element 13: North BL
        xpathResult = EvalXpath(xmlNav, tagArray(15))
        PopTextBoxByTag(13, xpathResult)

        '***** Element 14: South BL
        xpathResult = EvalXpath(xmlNav, tagArray(16))
        PopTextBoxByTag(14, xpathResult)
```

```
        '***** Element 15: Extent
        xpathResult = EvalXpath(xmlNav, tagArray(17))
'"metadata/dataIdInfo/descKeys[@KeyTypCd='002']/keyword")
        PopTextBoxByTag(15, xpathResult)

        '***** Element 16: Vertical Extent Information
        'Min Val
        xpathResult = EvalXpath(xmlNav, tagArray(18))
        PopTextBoxByTag(1601, xpathResult)
        'Max Val
        xpathResult = EvalXpath(xmlNav, tagArray(19))
        PopTextBoxByTag(1602, xpathResult)

        'Unit of Measure
        xpathResult = EvalXpath(xmlNav, tagArray(20))
        PopTextBoxByTag(1603, xpathResult)

        'Vertical Datum
        xpathResult = EvalXpath(xmlNav, tagArray(21))
        PopTextBoxByTag(1604, xpathResult)

        '***** Element 17: Spatial Reference System
        xpathResult = EvalXpath(xmlNav, tagArray(22))
        PopTextBoxByTag(17, xpathResult)

        '***** Element 18: Spatial Resolution
        xpathResult = EvalXpath(xmlNav, tagArray(23))
        PopTextBoxByTag(18, xpathResult)

        '***** Element 19: Spatial Representation Type
        xpathResult = EvalXpath(xmlNav, tagArray(24)) '& "/@value")
        PopTextBoxByTag(19, xpathResult)

        '***** Element 20: Presentation Type
        'Unknown
        'SimplePopTextBox(20, tagArray(25))

        '***** Element 21: Data Format
        xpathResult = EvalXpath(xmlNav, tagArray(26))
        PopTextBoxByTag(21, xpathResult)

        '***** Element 22: Supply Media
        xpathResult = EvalXpath(xmlNav, tagArray(27)) ' & "/@value")
        PopTextBoxByTag(22, xpathResult)

        '***** Element 23: Distributor
        'Distributor Name
        xpathResult = EvalXpath(xmlNav, tagArray(28))
        PopTextBoxByTag(2301, xpathResult)
        'Distributor POC
        xpathResult = EvalXpath(xmlNav, tagArray(29))
        PopTextBoxByTag(2302, xpathResult)

        'Distributor Address
        xpathResult = EvalXpath(xmlNav, tagArray(30))
        PopTextBoxByTag(2303, xpathResult)

        'Distributor City
        xpathResult = EvalXpath(xmlNav, tagArray(31))
        PopTextBoxByTag(2304, xpathResult)

        'Distributor Admin Area
        xpathResult = EvalXpath(xmlNav, tagArray(32))
        PopTextBoxByTag(2305, xpathResult)

        'Distributor PostCode
        xpathResult = EvalXpath(xmlNav, tagArray(33))
        PopTextBoxByTag(2306, xpathResult)

        'Distributor Country
        xpathResult = EvalXpath(xmlNav, tagArray(34))
        PopTextBoxByTag(2307, xpathResult)

        'Distributor Telephone
        xpathResult = EvalXpath(xmlNav, tagArray(35))
        PopTextBoxByTag(2308, xpathResult)
```

```
        'Distributor FAX
        xpathResult = EvalXpath(xmlNav, tagArray(36))
        PopTextBoxByTag(2309, xpathResult)

        'Distributor email
        xpathResult = EvalXpath(xmlNav, tagArray(37))
        PopTextBoxByTag(2310, xpathResult)

        'Distributor FAX
        xpathResult = EvalXpath(xmlNav, tagArray(38))
        PopTextBoxByTag(2311, xpathResult)

        '***** Element 24: Frequency of Update
        xpathResult = EvalXpath(xmlNav, tagArray(39))
'"metadata/dataIdInfo/resMaint/maintFreq/MaintFreqCd/@value")
        PopTextBoxByTag(24, xpathResult)

        '***** Element 25: Access Constraint
        xpathResult = EvalXpath(xmlNav, tagArray(40))
'"metadata/dataIdInfo/resConst/LegConsts/accessConsts/RestrictCd/@*")
        PopTextBoxByTag(25, xpathResult)

        '***** Element 26: Use Constraints
        xpathResult = EvalXpath(xmlNav, tagArray(41))
'"metadata/dataIdInfo/resConst/LegConsts/useConsts/RestrictCd/@*")
        PopTextBoxByTag(26, xpathResult)

        '***** Element 27: Additional Information Source
        'xpathResult = EvalXpath(xmlNav, tagArray(42)) - unknown element in ESRI-ISO
        'PopTextBoxByTag(27, xpathResult)

        '***** Element 28: Online Resource
        xpathResult = EvalXpath(xmlNav, tagArray(43))
        PopTextBoxByTag(28, xpathResult)

        '***** Element 29: Browse Graphic
        'xpathResult = EvalXpath(xmlNav, tagArray(44))
        'PopTextBoxByTag(29, xpathResult)

        '***** Element 30: Date of Update of Metadata
        xpathResult = EvalXpath(xmlNav, tagArray(45))
        PopTextBoxByTag(30, xpathResult)

        '***** Element 31: Metadata Standard Name
        xpathResult = EvalXpath(xmlNav, tagArray(46))
        PopTextBoxByTag(31, xpathResult)

        '***** Element 32: Metadata Standard Version
        xpathResult = EvalXpath(xmlNav, tagArray(47))
        PopTextBoxByTag(32, xpathResult)

End Function
Function EvalXpath(ByVal xmlNav As XPathNavigator, ByVal xpathString As String) As
String
        'Compile the XPath expression
        Dim xpathExpr As XPathExpression
        xpathExpr = xmlNav.Compile(xpathString)
        'Display the results depending on type of result
        Select Case (xpathExpr.ReturnType)

            Case XPathResultType.Boolean
                Dim res As Boolean
                'MsgBox("Boolean value: {0}" & xmlNav.Evaluate(xpathExpr))
                res = xmlNav.Evaluate(xpathExpr)
                Return res.ToString

            Case XPathResultType.String
                Dim res As String

                'MsgBox("String value: {0}" & xmlNav.Evaluate(xpathExpr))
                res = xmlNav.Evaluate(xpathExpr)
                Return res

            Case XPathResultType.Number
                Dim res As Integer
```

```
                'MsgBox("Number value: {0}" & xmlNav.Evaluate(xpathExpr))
                res = xmlNav.Evaluate(xpathExpr)
                Return res.ToString

            Case XPathResultType.NodeSet
                Dim xmlNodeIterator As XPathNodeIterator
                Dim val, val1 As String
                Dim int As Integer = 0

                xmlNodeIterator = xmlNav.Select(xpathString)
                While xmlNodeIterator.MoveNext() 'move through all nodes in current
result set
                    val = xmlNodeIterator.Current.Value
                    If (int = 0) Then
                        val1 = val1 & val
                    Else
                        val1 = val1 & ";" & val
                    End If
                    int += 1
                End While

                Return val1

            Case XPathResultType.Error
                MsgBox("XPath expression {0} is invalid." & xpathString)
        End Select
    End Function
    Function PopTextBoxByTag(ByVal txtBoxCnt As Integer, ByVal elementValue As
String)
        Dim txtIndex As String
        txtIndex = "TextBox" & txtBoxCnt
        FindControl(txtIndex, Me).Text = elementValue
End Function

Private Function FindControl(ByVal ControlName As String, ByVal CurrentControl As
Control) As Control
        For Each ctr As Control In CurrentControl.Controls
            If ctr.Name = ControlName Then
                Return ctr
            Else
                ctr = FindControl(ControlName, ctr)
                If Not ctr Is Nothing Then
                    Return ctr
                End If
            End If
        Next ctr
End Function
Function ShowThumb()
        Dim pGxThumb As IGxThumbnail
        pGxThumb = m_pGxDS

        Dim pictHandler As New ImageConverter
        'Me.PictureBox1.Image = pictHandler.IPictureToImage(pGxThumb.Thumbnail)
    End Function
    Function FindBlankFields(ByVal CurrentControl As Control) As ArrayList

        Dim emptyControls As New ArrayList
        For Each ctr As Control In CurrentControl.Controls
            If ctr.Text.Trim = "" And (TypeOf (ctr) Is TextBox) Then
                '             MsgBox(ctr.Name)
                emptyControls.Add(ctr.Name)
            End If
        Next
        Return emptyControls
End Function
Public Function GetFullPath(ByVal m_pGxDS As IGxDataset) As String
        Dim pWorkspace As IWorkspace
        pWorkspace = m_pGxDS.Dataset.Workspace
        Return pWorkspace.PathName
End Function
Function ReturnMinMaxDates(ByVal m_pGxDS As IGxDataset) As ArrayList
        Dim pDataset As IDataset
        Dim pFeatClass As IFeatureClass
        Dim pFeatws As IFeatureWorkspace
        Dim pFields As IFields
```

```
        Dim pField As IField
        Dim pTable As ITable
        Dim i As Integer
        Dim colArray As New ArrayList
        Dim dateArray As New ArrayList
        ' QI to get from IGxDataset (m_pGxDS) to enable access to the FC table
        pDataset = m_pGxDS.Dataset
        pFeatws = m_pGxDS.Dataset.Workspace
        pFeatClass = pFeatws.OpenFeatureClass(pDataset.Name)
        pTable = pFeatClass
        pFields = pFeatClass.Fields
        'Identify all columns of type date in FC & store in an array
        'date fields are type 5 according to ESRI
        For i = 0 To (pFields.FieldCount - 1)
            pField = pFields.Field(i)
            If (pField.Type = 5) Then
                colArray.Add(pField.Name)
            End If
        Next i
        Dim pCursor As ICursor
        Dim pRow As IRow
        Dim iIndex As Long
        Dim en As IEnumerator = colArray.GetEnumerator
        Dim dateMax As Date = #1/1/1000#
        Dim dateMin As Date = #12/31/9999#

        'Iterate through all date columns of the FC, returning a single min & max
value
        Try
            While en.MoveNext()
                pCursor = pTable.Search(Nothing, False)
                pRow = pCursor.NextRow
                iIndex = pRow.Fields.FindField(en.Current)
                Do While Not pRow Is Nothing
                    If (dateMax <= pRow.Value(iIndex)) Then
                        dateMax = pRow.Value(iIndex)
                    End If
                    If (dateMin >= pRow.Value(iIndex)) Then
                        dateMin = pRow.Value(iIndex)
                    End If
                    pRow = pCursor.NextRow
                Loop
            End While

            pCursor = Nothing
            pRow = Nothing

            dateArray.Add(dateMin)
            dateArray.Add(dateMax)
            Return dateArray
        Catch
            MsgBox("No date fields found in attribute table")
        End Try

    End Function
    Function GetVerticalExtentInfo(ByVal m_pGxDS As IGxDataset) As ArrayList

        Dim pDataset As IDataset
        Dim pFeatClass As IFeatureClass
        Dim pFeatws As IFeatureWorkspace
        Dim pFields As IFields
        Dim pField As IField
        Dim pTable As ITable
        Dim pGeoDef As IGeometryDef
        Dim lGeomIndex As Long

        'Set the Z domain extent for the dataset. The datatype for the
        'numeric expressions that represent inZMin and inZMax must be double.

        ' QI to get from IGxDataset (m_pGxDS) to enable access to the Feature Class
Table
        pDataset = m_pGxDS.Dataset
        pFeatws = m_pGxDS.Dataset.Workspace
        pFeatClass = pFeatws.OpenFeatureClass(pDataset.Name)
        pTable = pFeatClass
        pFields = pFeatClass.Fields
```

```
        'test if the FC has Z defined - select the geometry (Shape) field
        lGeomIndex = pFields.FindField("Shape")
        pField = pFields.Field(lGeomIndex)
        pGeoDef = pField.GeometryDef

        'Test if the Feature class hasZ, pull out the Z values from the feature
dataset if true.
        If pGeoDef.HasZ = True Then

            Dim pGDS As IGeoDataset
            Dim dZmin As Double
            Dim dZmax As Double
            Dim zUnit As String
            'Uncomment if zUnits (Precision of Z) is required
            'Dim falseZ As Double
            'Dim zUnits As Double

            pGDS = pDataset
            If TypeOf pGDS.SpatialReference Is IProjectedCoordinateSystem Then
                Dim pPCS As IProjectedCoordinateSystem
                pPCS = pGDS.SpatialReference

                'Uncomment if zUnits (Precision of Z) is required
                'pPCS.GetZFalseOriginAndUnits(falseZ, zUnits)

                'Get the max & min z values
                pPCS.GetZDomain(dZmin, dZmax)

                'Get the units of measure. ZCoordinateUnit does not appear stable,
defaulting to Coord system's unit
                zUnit = pPCS.CoordinateUnit.Name
                MsgBox(zUnit)
            End If

            If TypeOf pGDS.SpatialReference Is IGeographicCoordinateSystem Then
                Dim pGCS As IGeographicCoordinateSystem
                pGCS = pGDS.SpatialReference

                'Uncomment if zUnits (Precision of Z) is required
                'pGCS.GetZFalseOriginAndUnits(falseZ, zUnits)

                'Get the max & min z values
                pGCS.GetZDomain(dZmin, dZmax)

                'Get the units of measure. ZCoordinateUnit does not appear stable,
defaulting to Coord system's unit
                zUnit = pGCS.CoordinateUnit.Name
                MsgBox(zUnit)
            End If

            Dim vertExArray As New ArrayList
            vertExArray.Add(dZmin)
            vertExArray.Add(dZmax)
            vertExArray.Add(zUnit)
            'Uncomment if zUnits (Precision of Z) is required
            'vertExArray.Add(falseZ)

            Return vertExArray
        End If

        Return Nothing

End Function
Function GetSpatialResolution() As Double

        Dim pDataset As IDataset
        Dim pGDS As IGeoDataset
        Dim falseX, falseY, xyUnits As Double
        pDataset = m_pGxDS.Dataset
        pGDS = pDataset

        If TypeOf pGDS.SpatialReference Is IProjectedCoordinateSystem Then
            Dim pPCS As IProjectedCoordinateSystem

            pPCS = pGDS.SpatialReference
```

```
            pPCS.GetFalseOriginAndUnits(falseX, falseY, xyUnits)
        End If

        If TypeOf pGDS.SpatialReference Is IGeographicCoordinateSystem Then
            Dim pGCS As IGeographicCoordinateSystem

            pGCS = pGDS.SpatialReference
            pGCS.GetFalseOriginAndUnits(falseX, falseY, xyUnits)
        End If
        Return xyUnits
    End Function

Public Function CurrentUser() As String
        Dim username As String
        username = SystemInformation.UserName
        Return username
End Function
Function GetSystemLanguage()
        Dim objectQuery As New WqlObjectQuery("select * from win32_OperatingSystem")
        Dim searcher As New ManagementObjectSearcher(objectQuery)
        Dim share As ManagementObject
        Dim a As String
        Dim os As String

        For Each share In searcher.Get()
            a = share("Name")
            Dim split1 As String()
            split1 = Split(a, "|")
            os = split1(0)
            MsgBox(os)
        Next share

        For Each share In searcher.Get()
            Dim a1 As String
            a1 = Convert.ToString(share("OSLanguage"))
            MsgBox(a1)
            If a1 = "0409" Or "1033" Then
                MsgBox("English")
            ElseIf a1 = "0407" Then
                MsgBox("German")
            ElseIf a1 = "040a" Then
                MsgBox("Spanish")
            End If
        Next
End Function

Function RegionalSettings()
        Dim ci As CultureInfo
        Dim str As String
        str = ci.CurrentCulture.ThreeLetterISOLanguageName
End Function

Function ElementsFromDirTree()
       ' =*=* get file path of current layer
        Dim dirs As String() = Nothing
        Dim pos As Integer
        Dim dirArray As ArrayList

        'Read full filepath from csv
        Try
            dirArray = ReadTagsFromFile(geminiFileTree, 1)
        Catch
            MsgBox("no file found")
        End Try

        Dim root, dirPath As String
        'Start at 2 as folder elements start here in CSV
        Dim eleNo As Integer = 2

        'Root of the data tree as specified in the csv
        root = dirArray(1)
        MsgBox(root)

        'Slice off root from the full path of the FC
        pos = root.Length
        dirPath = GetFullPath(m_pGxDS).Substring(pos)
```

```
        'slice off the .mdb from the path tail
        dirPath = dirPath.Remove((dirPath.Length - 4), 4)

        'get the feature dataset name
        Dim pDataset As IDataset
        Dim pFeatws As IFeatureWorkspace
        Dim pFeatClass As IFeatureClass

        pDataset = m_pGxDS.Dataset
        pFeatws = m_pGxDS.Dataset.Workspace
        pFeatClass = pFeatws.OpenFeatureClass(pDataset.Name)

        'add feature dataset name to the string
        dirPath = dirPath & "\" & pFeatClass.FeatureDataset.Name

        'split remaining path components giving metadata elements
        dirs = dirPath.Split("\")

        'Move through the string array and populate textboxes on the basis of indexes
specified in .csv file
        Dim en As IEnumerator = dirs.GetEnumerator
        While en.MoveNext
            'For eact
            MsgBox("element " & en.Current)
            PopTextBoxByTag(dirArray(eleNo), en.Current)
            eleNo += 1
        End While

End Function

Function GetAlternativeTitle() As String
       'get the feature dataset name
        Dim pDataset As IDataset
        Dim pFeatws As IFeatureWorkspace
        Dim pFeatClass As IFeatureClass
        pDataset = m_pGxDS.Dataset
        pFeatws = m_pGxDS.Dataset.Workspace
        pFeatClass = pFeatws.OpenFeatureClass(pDataset.Name)
        Return pFeatClass.AliasName
End Function

Private Sub btnGEMINI_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnGEMINI.Click

        ExportGEMINI()
        'ExportFGDC()
        'ValidateXml()
        'UpdateESRIXML()
    End Sub

Function UpdateESRIXML()
        'Dim pPropSet As IPropertySet
        ''changes the abstract
        'm_xmlprops.SetPropertyX("dataIdInfo/idAbs", Me.TextBox4.Text,
esriXmlPropertyType.esriXPTText, esriXmlSetPropertyAction.esriXSPAAddOrReplace,
False)
        'pPropSet = m_xmlprops
        ''saves the metadata
        'm_pMetaData.Metadata = pPropSet
    End Function
Public Shared Sub ValidateXml()
        Dim stream As New
System.IO.FileStream("D:\vbnet\textFolder\output\temp_gemini.xml", FileMode.Open)
        Dim vr As New XmlValidatingReader(stream, XmlNodeType.Element, Nothing)
        vr.Schemas.Add(Nothing,
"D:\vbnet\textFolder\schemas\gemini\geminiSchema.xsd")
        vr.Schemas.Add(Nothing, "D:\vbnet\textFolder\schemas\gemini\gcoGemini.xsd")
        vr.Schemas.Add(Nothing, "D:\vbnet\textFolder\schemas\gemini\gmdGemini.xsd")
        vr.Schemas.Add(Nothing, "D:\vbnet\textFolder\schemas\gemini\gmlGemini.xsd")
        vr.Schemas.Add(Nothing, "D:\vbnet\textFolder\schemas\gemini\gtsGemini.xsd")
        vr.Schemas.Add(Nothing, "D:\vbnet\textFolder\schemas\gco\gco.xsd")
        vr.Schemas.Add(Nothing, "D:\vbnet\textFolder\schemas\gmd\gmd.xsd")
        vr.Schemas.Add(Nothing, "D:\vbnet\textFolder\schemas\gml\base\gml.xsd")
        vr.Schemas.Add(Nothing, "D:\vbnet\textFolder\schemas\gml\smil\smil20.xsd")
        vr.Schemas.Add(Nothing, "D:\vbnet\textFolder\schemas\gml\xlink\xlinks.xsd")
```

```
        vr.Schemas.Add(Nothing, "D:\vbnet\textFolder\schemas\gmx\gmx.xsd")
        vr.Schemas.Add(Nothing, "D:\vbnet\textFolder\schemas\gsr\gsr.xsd")
        vr.Schemas.Add(Nothing, "D:\vbnet\textFolder\schemas\gss\gss.xsd")
        vr.Schemas.Add(Nothing, "D:\vbnet\textFolder\schemas\gts\gts.xsd")
        vr.Schemas.Add(Nothing, "D:\vbnet\textFolder\schemas\gts\gts.xsd")

        vr.ValidationType = ValidationType.Schema
        AddHandler vr.ValidationEventHandler, AddressOf ValidationHandler

        While vr.Read()
        End While
        MsgBox("Validation finished")
    End Sub
Public Shared Sub ValidationHandler(ByVal sender As Object, ByVal args As
ValidationEventArgs)
        MsgBox("***Validation error")
        MsgBox("Severity:{0}", args.Severity)
        MsgBox("Message:{0}", args.Message)
    End Sub
End Class
```

Appendix E – Geosemantic service prototype code

index.html

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN" "DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
    <head>
     <title>OpenLayers map preview</title>
     <style type="text/css">
      #map {
        width: 600px;
        height: 400px;
        border: 1px solid black;
      }
     </style>
<script src="http://localhost:8080/geoserver/openlayers/OpenLayers.js"
type="text/javascript"></script>
     <script defer="defer" type="text/javascript">
       var map;
       var untiled;
       var tiled;
       function setHTML(response) {
        OpenLayers.Util.getElement('nodelist').innerHTML = response.responseText;
       };

       function MsgBox(textstring){
       alert(textstring)
       };

       function getPix(pixel){
       var lonlat = map.getLonLatFromPixel(pixel);
       alert ("Lon: "+lonlat.lon + " (Pixel.x: " + pixel.x +")" + "\n" + "Lon:
"+lonlat.lat + "(Pixel.y: " + pixel.y +")" );

       document.forms.Form1.xVal.value = lonlat.lon;
       document.forms.Form1.yVal.value = lonlat.lat;
       };

       function getScreenDimension(){
       //alert("test");
       alert("Java Version: " + java.lang.System.getProperty("java.vendor"));
       };

       function init(){
          map = new OpenLayers.Map('map', {controls:[], 'projection': 'EPSG:27700',
'units':'m'});

          OpenLayers.IMAGE_RELOAD_ATTEMPTS = 5;

          // setup tiled layer
          var bounds = new
OpenLayers.Bounds(324900.1054434324,672087.5954114769,327097.6495371759,674193.752249
4969);
          tiled = new OpenLayers.Layer.WMS(
            "topp:edinburgh", "http://localhost:8080/geoserver/wms",
            {
              layers: 'topp:edinburgh',
              styles: '',
              height: '719',
              width: '800',
              srs: 'EPSG:27700',
              format: 'image/png', tiled: 'true', tilesOrigin :
"324900.1054434324,672087.5954114769"
            },
            {maxExtent: bounds, maxResolution: 8.58415661618551, projection:
"EPSG:27700", buffer: 0}
          );
          //map.addLayer(tiled);
```

```
          // setup untiled layer
          untiled = new OpenLayers.Layer.WMS.Untiled(
            "topp:edinburgh", "http://localhost:8080/geoserver/wms",
            {
              layers: 'topp:edinburgh',
              styles: '',
              height: '719',
              width: '800',
              srs: 'EPSG:27700',
              format: 'image/png'
            },
            {maxExtent: bounds, maxResolution: 8.58415661618551, projection:
"EPSG:27700"}
          );
          untiled.ratio=1;
          untiled.setVisibility(false, false);
          map.addLayer(untiled);

          // setup controls and initial zooms
          map.addControl(new OpenLayers.Control.PanZoomBar({div:$('nav')}));
          map.addControl(new OpenLayers.Control.MouseDefaults());
          map.addControl(new OpenLayers.Control.Scale($('scale')));
          map.addControl(new OpenLayers.Control.MousePosition({element:
$('position')}));
          //map.addControl(new OpenLayers.Control.LayerSwitcher());
          //map.addControl(new OpenLayers.Control.OverviewMap());
          map.zoomToExtent(bounds);

          // support GetFeatureInfo
          map.events.register('click', map, function (e) {
      OpenLayers.Util.getElement('nodelist').innerHTML = "Loading... please wait...";
            var url =  map.layers[0].getFullRequestString({
                            REQUEST: "GetFeatureInfo",
                            EXCEPTIONS: "application/vnd.ogc.se_xml",
                            BBOX: map.getExtent().toBBOX(),
                            X: e.xy.x,
                            Y: e.xy.y,
                            INFO_FORMAT: 'text/html',
                            QUERY_LAYERS: map.layers[0].params.LAYERS,
                            FEATURE_COUNT: 50,
                            layers: 'topp:edinburgh',
                            styles: '',
                            srs: 'EPSG:27700',
                            WIDTH: map.size.w,
                            HEIGHT: map.size.h},
                            "http://localhost:8080/geoserver/wms"
                            );
            OpenLayers.loadURL(url, '', this, setHTML, setHTML);

            var pixel = new OpenLayers.Pixel(e.xy.x,e.xy.y);
            getPix(pixel);
            Event.stop(e);
      });
      }
</script>
  </head>
  <body onload="init()">
     <table>
       <tr>
         <td style="width:40px" valign="middle" rowspan="3"><div id="nav"></div></td>
         <td colspan="3" align="right">
           <!-- Switch layers when links are pressed -->
           <a id="untiledLink" href="#"
onclick="map.removeLayer(tiled);map.addLayer(untiled)">Untiled</a>
           <a id="tiledLink" href="#"
onclick="map.removeLayer(untiled);map.addLayer(tiled);">Tiled</a>
         </td>
       </tr>
       <tr>
         <td colspan="3"><div id="map"></div></td>
       </tr>
       <tr>
         <td><div id="scale"></div></td>
         <td/>
         <td align="right"><div id="position"></div></td>
       </tr>
```

```
    </table>
    This is a <a href="Qservlet"> link to the QM servlet</a>
    <form name="Form1" method="post" action="Qservlet">
      <input type="hidden" name="xVal">
      <input type="hidden" name="yVal">
      <input type="text" name="buffer">
      <input type="submit" value="Submit">
    </form>
    <div id="nodelist">Click on the map to get feature infos</div>
  </body>
</html>
```

map.js

```
<script src="http://localhost:8080/geoserver/openlayers/OpenLayers.js"></script>
    <script type="text/javascript">
        var map, layer;
        function init(){
map = new OpenLayers.Map('map');
//         OpenLayers.IMAGE_RELOAD_ATTEMPTS = 5;

          var bounds = new
OpenLayers.Bounds(324900.1054434324,672087.5954114769,327097.6495371759,674193.752249
4969);

          // setup untiled layer
          layer = new OpenLayers.Layer.WMS.Untiled(
            "topp:edinburgh", "http://localhost:8080/geoserver/wms",
            { layers: 'topp:edinburgh',
            styles: '',
              height: '719',
              width: '800',
              srs: 'EPSG:27700',
              format: 'image/png'},{maxExtent: bounds, maxResolution:
8.58415661618551, projection: "EPSG:27700"} );
          map.addLayer(layer);

      //Markers portion
            var markers = new OpenLayers.Layer.Markers( "Markers",{maxExtent: bounds,
maxResolution: 8.58415661618551, projection: "EPSG:27700"} );
            map.addLayer(markers);

            var size = new OpenLayers.Size(10,17);
            var offset = new OpenLayers.Pixel(-(size.w/2), -size.h);
            var icon = new
OpenLayers.Icon('http://localhost:8080/jkb_proto/AQUA.png',size,offset);
            var i=0;
            markers.addMarker(new OpenLayers.Marker(new
OpenLayers.LonLat(centXArray[0],centYArray[0]),icon));
            var iclone = icon.clone();

            //markers.addMarker(new OpenLayers.Marker(new
OpenLayers.LonLat(centXArray[1],centYArray[1]),iclone));

            for (i=1; i<=centXArray.length;i++){
               var iclone = icon.clone();
               markers.addMarker(new OpenLayers.Marker(new
OpenLayers.LonLat(centXArray[i],centYArray[i]),iclone));
            }


            //map.addControl(new OpenLayers.Control.LayerSwitcher());
            map.zoomToMaxExtent();

            halfIcon.setOpacity(0.5);
      //end markers portion
}
    </script>
```

head.html

```
<HTML>
<HEAD>
<title>JavaScriptExample</title>
     <style type="text/css">
      #map {
        width: 600px;
        height: 400px;
        border: 1px solid black;
      }
     </style>
```

foot.html

```
</head>
  <body onload="init()">
     <table>
       <tr>
         <td style="width:40px" valign="middle" rowspan="3"><div id="nav"></div></td>
         <td colspan="3" align="right">
           <!-- Switch layers when links are pressed -->
           <a id="untiledLink" href="#"
onclick="map.removeLayer(tiled);map.addLayer(untiled)">Untiled</a>
           <a id="tiledLink" href="#"
onclick="map.removeLayer(untiled);map.addLayer(tiled);">Tiled</a>
         </td>
       </tr>
       <tr>
         <td colspan="3"><div id="map"></div></td>
       </tr>
       <tr>
         <td><div id="scale"></div></td>
         <td/>
         <td align="right"><div id="position"></div></td>
       </tr>
     </table>
     This is a <a href="index.html"> link to the QM servlet</a>
     <div id="nodelist">Click on the map to get feature infos</div>
  </body>
</html>
```

Qservlet.java

```
package uk.ac.ed;

import java.io.IOException;
import java.io.InputStream;
import java.io.PrintWriter;
import java.net.URL;
import java.sql.SQLException;
import java.util.Enumeration;
import java.util.LinkedList;
import java.util.List;
import java.util.Vector;
import javax.servlet.RequestDispatcher;
import javax.servlet.ServletContext;
import javax.servlet.ServletException;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import com.hp.hpl.jena.graph.Triple;

/**
 * Servlet implementation class for Servlet: Qservlet
 *
 */
public class Qservlet extends javax.servlet.http.HttpServlet implements
javax.servlet.Servlet {
static final long serialVersionUID = 1L;
```

```
    /* (non-Java-doc)
         * @see javax.servlet.http.HttpServlet#HttpServlet()
         */
public Qservlet() {
super();
        }

        /* (non-Java-doc)
         * @see javax.servlet.http.HttpServlet#doGet(HttpServletRequest request,
HttpServletResponse response)
         */
protected void doGet(HttpServletRequest request, HttpServletResponse response) throws
ServletException, IOException {
                // TODO Auto-generated method stub
        }

        /* (non-Java-doc)
         * @see javax.servlet.http.HttpServlet#doPost(HttpServletRequest request,
HttpServletResponse response)
         */
protected void doPost(HttpServletRequest request, HttpServletResponse response)
throws ServletException, IOException {
// TODO Auto-generated method stub
                response.setContentType("text/html");
                PrintWriter out = response.getWriter();

//              out.println("doPost entered");
//Convert the text parameters to Doubles for passing to the QProcessor class
//              Double xCoord = new Double (request.getParameter("xVal"));
                String xCoord = request.getParameter("xVal");
//              xCoord = xCoord.doubleValue();
//              Double yCoord = new Double (request.getParameter("yVal"));
                String yCoord = request.getParameter("yVal");
//              yCoord = yCoord.doubleValue();

//              Double buff = new Double (request.getParameter("buffer"));
                String buff = request.getParameter("buffer");
//              buff = buff.doubleValue();
                Double testX = new Double(324700);
                Double testY = new Double(673000);
                LinkedList<Integer> gid;
//              gid = new LinkedList();
                LinkedList<Double> centroidX;
//              centroidX = new LinkedList();
//              LinkedList<Double> centroidY;
//              centroidY = new LinkedList();

                //Double [] centX;
                int size = -1;
                TripleProcessor tp = new TripleProcessor();
                try{
                        Triple triple=null;
                        try {
                                triple = tp.TripleProcessor(null,null, "rdf:type",
null, null );
                        } catch (Exception e1) {
                                // TODO Auto-generated catch block
                                e1.printStackTrace();
                        }

                        //Instantiate point
                        MyPoint point = new MyPoint(xCoord, yCoord);
                        //Convert buffer value to double & add
                        Double buffer = new Double (buff);
                        buffer = buffer.doubleValue();
                        QP2 qp = new QP2();

//                      Vector rsVector = qp.QueryProcessor(xCoord, yCoord, buff);
Point, buffer, triple, sparql
                        Vector rsVector = qp.QP2(point, buffer, triple, null);
                        Enumeration en = rsVector.elements();
                        size = rsVector.size();

                        if(rsVector.isEmpty()){
                                out.println("empty Result set");
                        }
```

```
                        else{
                                ServletContext sc = this.getServletContext();

        //out.println("<HTML><HEAD><title>JavaScriptExample</title>");
                                RequestDispatcher rd =
sc.getRequestDispatcher("/head.html");
                                if (rd!=null){
                                        try{
                                                rd.include(request, response);
                                                out.print("Head include OK");
                                        }
                                        catch (Exception e) {
                                                System.out.println("include problem with
head: " + e.getMessage());
                                        }

                                //Code to populate a javascript array with servlet
variables
                                out.println("<script language=JavaScript>");
                                out.println("var centXArray = new Array(");
                                Boolean isFirst = true;
                                //Iterate though array
                                for (int i=0; i< size; i=i+3){
                                        if(!isFirst){
                                                out.print(",");
                                        }
                                        isFirst=false;
//                                      out.print("var centX = ");
                                        out.print(rsVector.get(i));
                                }
                                out.println(");");

                                out.println("var centYArray = new Array(");
                                Boolean isFirst2 = true;
                                //Iterate though array
                                for (int i=1; i< size; i=i+3){
                                        if(!isFirst2){
                                                out.print(",");
                                        }
                                        isFirst2=false;
//                                      out.print("var centX = ");
                                        out.print(rsVector.get(i));
                                }
                                out.println(");");
                                out.println("</script>");
                                }

                                //rd = sc.getRequestDispatcher("/result.html");
                                rd = sc.getRequestDispatcher("/map.js");
                                if (rd!=null){
                                        try{
                                                rd.include(request, response);
                                                out.print("js include OK");
                                        }
                                        catch (Exception e) {
                                                System.out.println("include problem with
javascript: " + e.getMessage());
                                        }
                                }

                                rd = sc.getRequestDispatcher("/foot.html");
                                if (rd!=null){
                                        try{
                                                rd.include(request, response);
                                                out.print("Foot include OK");
                                        }
                                        catch (Exception e) {
                                                System.out.println("include problem with
foot: " + e.getMessage());
                                        }
                                }

                                out.close();
                        }
```

```
                }
                catch (SQLException e) {
                        // TODO Auto-generated catch block
                        out.println("PROBLEM!!!");
                        e.printStackTrace();
                }
 }
}
```

TripleProcessor.java

```
package uk.ac.ed;

import java.util.Vector;
import com.hp.hpl.jena.graph.Node;
import com.hp.hpl.jena.graph.Triple;
import com.hp.hpl.jena.ontology.DatatypeProperty;
import com.hp.hpl.jena.ontology.ObjectProperty;
import com.hp.hpl.jena.ontology.OntClass;
import com.hp.hpl.jena.rdf.model.Property;
import com.hp.hpl.jena.rdf.model.Resource;
import com.hp.hpl.jena.vocabulary.OWL;
import com.hp.hpl.jena.vocabulary.RDF;
import com.hp.hpl.jena.vocabulary.RDFS;

public class TripleProcessor {
        public static String ontologyNS = "http://localhost/edbuildings.owl#";
        public static String instanceNS = "http://eeo.ed.ac.uk/";

//      public Vector UIProcessor(String xCoord, String yCoord, String buff, String s,
String sTest, String p, String o, String oTest) throws Exception{
        public Triple TripleProcessor(String s, String sTest, String p, String o,
String oTest) throws Exception{

                System.out.println("UI Entered OK");
                //Test s, p, o & prepend the appropriate namespace
                Node subject, predicate, object;
                if (s !=null){
                        subject = prependNS(s, sTest);
                        System.out.println("Subject " + subject);
                }
                else {
                        subject = Node.ANY;
                }
                if (p != null){
                        predicate = prependNS(p, "1");
                        System.out.println("Predicate " + predicate);
                }
                else{
                        predicate = Node.ANY;
                }

                if (s != null){
                        object = prependNS(o, oTest);
                        System.out.println("Object " + object);
                }
                else{
                        object = Node.ANY;
                }

                Triple pattern = new Triple(subject, predicate, object);
                return pattern;
        }

        private Node prependNS(String str, String resTest) throws Exception{
        //Helper method to prepend NS

                //Node testN = Node.create("http://localhost/edbuildings.owl#Bar");
                Node testN;

                if(resTest=="1"){
                        str = this.ontologyNS+str;
                        testN = Node.create(str);
```

```
                }
                else{
                        str = this.instanceNS+str;
                        testN = Node.create(str);
                }
                return testN;
        }
}
```

MyPoint.java

```
package uk.ac.ed;

public class MyPoint {
        //Instance variables
        public static double xc;
        public static double yc;
        public static String id;
        public MyPoint(String xCoord, String yCoord) {

                //System.out.println("Point entered...");

                //Convert the text parameters to Doubles for passing to the QProcessor
class
                //this.xc = convertStringToDouble(xCoord);
                setX(xCoord);

                //this.yc = convertStringToDouble(yCoord);
                setY(yCoord);

                this.id=null;

        }

        public MyPoint(String xCoord, String yCoord, String id) throws Exception{

                //Convert the text parameters to Doubles for passing to the QProcessor
class
                //this.xc = convertStringToDouble(xCoord);
                setX(xCoord);

                //this.yc = convertStringToDouble(yCoord);
                setY(yCoord);
                setID(id);
        }

        public MyPoint() {
                // TODO Auto-generated constructor stub
        }

        public double getX(){
                return this.xc;
        }

        public double getY(){
                return this.yc;
        }

        public String getID(){
                return id;
        }

        public String pointsToString(){
                String ptString = this.xc + " " + this.yc;
                return ptString;
        }

        public void setX(String xCoord){
                this.xc = convertStringToDouble(xCoord);
        }

        public void setY(String yCoord){
                this.yc = convertStringToDouble(yCoord);
```

```
        }

        public void setID(String id){
                this.id = id;
        }

        private Double convertStringToDouble(String coord){
                Double tempCoord = new Double(coord);
                tempCoord = tempCoord.doubleValue();
                return tempCoord;
        }
}
```

QP2.java

```
package uk.ac.ed;

import java.sql.Array;
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.SQLException;
import java.util.ArrayList;
import java.util.Iterator;
import java.util.LinkedList;
import java.util.List;
import java.util.Vector;
import org.mindswap.pellet.jena.PelletReasonerFactory;
import com.hp.hpl.jena.graph.Node;
import com.hp.hpl.jena.graph.Triple;
import com.hp.hpl.jena.graph.impl.GraphBase;
import com.hp.hpl.jena.ontology.OntClass;
import com.hp.hpl.jena.ontology.OntModel;
import com.hp.hpl.jena.query.Query;
import com.hp.hpl.jena.query.QueryExecutionFactory;
import com.hp.hpl.jena.query.QueryFactory;
import com.hp.hpl.jena.query.QuerySolution;
import com.hp.hpl.jena.query.ResultSet;
import com.hp.hpl.jena.rdf.model.InfModel;
import com.hp.hpl.jena.rdf.model.Model;
import com.hp.hpl.jena.rdf.model.ModelFactory;
import com.hp.hpl.jena.rdf.model.Property;
import com.hp.hpl.jena.rdf.model.Resource;
import com.hp.hpl.jena.rdf.model.Statement;
import com.hp.hpl.jena.rdf.model.StmtIterator;
import com.hp.hpl.jena.reasoner.Reasoner;
import com.hp.hpl.jena.reasoner.ReasonerRegistry;
import com.hp.hpl.jena.reasoner.ValidityReport;
import com.hp.hpl.jena.util.FileManager;
import com.hp.hpl.jena.util.PrintUtil;
import com.hp.hpl.jena.vocabulary.OWL;
import com.hp.hpl.jena.vocabulary.RDF;
import com.hp.hpl.jena.vocabulary.RDFS;
import de.fuberlin.wiwiss.d2rq.GraphD2RQ;
import de.fuberlin.wiwiss.d2rq.ModelD2RQ;
import de.fuberlin.wiwiss.d2rq.sql.ConnectedDB;

public class QP2 extends Object{
//public java.sql.ResultSet class QProcessor (Double x) extends Object{

        //public static double pointX = 326085;
        //public static double pointY = 673313;
        //public static double buffer = 500;
        public static Resource subCondition = null;
        public static Property predCondition = null;
        public static Resource objCondition = null;
        public static String sparqlCondition = null;
//      public static java.sql.ResultSet dbRS = null;

        public static List list;
        public static String attributeClass =
"http://localhost/edbuildingsp.owl#PublicHouses";
        public static String subClass = "http://localhost/edbuildings.owl#Bar";
        public static String mapURL = "/java_ws/edconf/ed_map.n3";
```

```
        public static String ontURL = "file:/java_ws/edconf/edbuildingsp.owl";
        public static String dbURL = "jdbc:postgresql://localhost:5432/pg_db_sp";
        public static String dbUsername = "pguser";
        public static String dbPassword = "pgpwd";


//      public static void main (String args[]) throws Exception{
//      public Vector QP2(Double x, Double y, Double buff, String s, String p, String
o, String sparql) throws SQLException{

        public Vector QP2(MyPoint pt, Double buffer, Triple trip, String sparql)
throws SQLException{
                //Return a vector filled with centroid xCoord, centroid yCoord and gid

                //Initialise constructor variables
//              this.pointX = x;
////            this.pointY = y;
//              this.buffer = buff;
//              this.subCondition = s;
//              this.predCondition = p;
//              this.objCondition = o;
//              this.sparqlCondition = sparql;

                //Point, buffer, Triple, Sparql
                Node as = Node.create("http://localhost/edbuildings.owl#Bar");
                Node as1 = Node.ANY;
                //Triple trip = new Triple(temp, RDF.type.asNode(), as1);
                //MyPoint pt = new MyPoint("326085", "673313");
                //Double buffer = 10000.00;
                //Create inference model from ontology and instance data (virtualised
using D2RQ)
                InfModel inf = initD2RQOWL(mapURL, ontURL);
                //InfModel inf = initD2RQPellet(mapURL, ontURL);

                //Test UIProcessor - code to be moved...
                //TripleProcessor ui = new TripleProcessor();
                //String ptx = "326085";
                //String pty = "673313";
                //Triple trip = ui.TripleProcessor (null,null, "hasAlcoholLicense",
null, null );

                //System.out.println("Vector created OK");
                //MyPoint pt = (MyPoint) v.get(0);
                Double xc = pt.getX();
                System.out.println(xc);
                Double yc = pt.getY();
                System.out.println(yc);
//              Double buffer = (Double) v.get(1);
                System.out.println("Buffer " + buffer);
                Node subj = trip.getSubject();
                System.out.println("FAIL");

                //String subj = (String) v.get(2);
                Resource subject = null;
                if(subj != Node.ANY){
                        System.out.println("Subject " + subj.toString());
                        subject = inf.getResource(subj.toString());
                }
                else{
                        subject = null;
                }
                Node pred = trip.getPredicate();
                Property predicate = null;
                if(pred != Node.ANY){
                        System.out.println("Predicate " + pred.toString());
                        predicate = inf.getProperty(pred.toString());
                }
                else{
                        predicate = null;
                }

                //System.out.println(pred.toString());
                Node obj = trip.getObject();
                Resource object = null;

                if(obj != Node.ANY){
```

```
                        System.out.println("Object "+ obj.toString());
                        object = inf.getResource(obj.toString());
                }
                else{
                        object = null;
                }

                //Test whether a valid inference can be made
                String testCondition = testInfer(inf,subject, predicate, object);

                if (testCondition == "true"){
                        System.out.println("All is Cool and the Gang");
                }
                else{
                        System.out.println("No corresponding statements exist");
                }

                //printStatements(inf, subject, predicate, object);
                //Create a model holding all result set statements
                Model tmpResModel = createResultModel(inf, subject, predicate,
object);
                //tmpResModel.write(System.out, "RDF/XML");
                //tmpResModel.write(System.out, "N3");
                //Close inference model
                //inf.close();
                System.out.println("Pred " + trip.getPredicate().toString());
                //Assign SPARQL query parameters
                //String subCond = "jkb:Building";
                String subCond = "?s";
                //String subCond = subj.toString();
                //String predCond = "jkb:hasAlcoholLicense";
                String predCond = "rdf:type";
                //String predCond = trip.getPredicate().toString();
                //String predCond = "?p";
                //String objCond = "?o";
                String objCond = "jkb:Pub";
                //System.out.println("Obj " + trip.getObject().toString());
                //Run Sparql query on the (temporary) result model
                //ResultSet rdfRS = runSimpleSparql(tmpResModel, subCond, predCond,
objCond);
                ResultSet rdfRS2 = runSimpleSparql(inf, subCond, predCond, objCond);
                //Parse sparql query to return a list of attributes for query
                //This needs to be extended to deal with other scenarios
                list = parseSparqlForGID(rdfRS2);
                //Close the temporary model
                tmpResModel.close();
                //Connect to DB for conventional query
                Connection dbConn = dbConnect(dbURL, dbUsername, dbPassword);
                //Construct database query
//              Vector vRS = queryDB(list, dbConn, buffer, pt);
                Vector vRS = null;

                if (list.size()==0){
                        System.out.println("Empty result set in sparql query");
                }
                else{
                        vRS = queryDB(list, dbConn, buffer, pt);
                        if(vRS.isEmpty())
                        {
                                System.out.println("empty Result set in QP");
                        }
                        else{
                                System.out.println("Non-empty Result set in QP");
                                }

//                              while (dbRS.next()){

//                                      System.out.println(dbRS.getString(1));
                }
                        //dbRS.first();
                return vRS;
        }
        public static InfModel initD2RQOWL(String mapURL, String ontURL) {
                //Set up the ModelD2RQ using a mapping file & apply an appropriate
namespace
                Model data = new ModelD2RQ("file:"+ mapURL, null,
```

```
"http://eeo.ed.ac.uk/");
                Model schema = FileManager.get().loadModel(ontURL);
                //use MiniReasoner as full is somewhat slow (hangs the code)
                Reasoner reasoner = ReasonerRegistry.getOWLMiniReasoner();
                reasoner = reasoner.bindSchema(schema);
                InfModel infmodel = ModelFactory.createInfModel(reasoner, data);
                //infmodel.write(System.out, "RDF/XML");
                //infmodel.write(System.out);
                return infmodel;
        }

        public static InfModel initD2RQPellet(String MapURL, String ontURL){
                //Pellet attempt
                //OntModel mod =
ModelFactory.createOntologyModel(PelletReasonerFactory.THE_SPEC);
                 System.out.println("Results with OntModel");
                 System.out.println("---------------------");
                 System.out.println();
                 // ontology that will be used
                 //String ont = ontURL;
                 Model data = new ModelD2RQ("file:"+ mapURL, null,
"http://eeo.ed.ac.uk/");
                 Model schema = FileManager.get().loadModel(ontURL);
                 // create an empty ontology model using Pellet spec
                 //OntModel model = ModelFactory.createOntologyModel(
PelletReasonerFactory.THE_SPEC );
                 Reasoner reasoner = PelletReasonerFactory.THE_SPEC.getReasoner();
                 reasoner = reasoner.bindSchema(schema);
                 InfModel inf = ModelFactory.createInfModel(reasoner, data);
                 //Reasoner reasoner = PelletReasonerFactory.THE_SPEC;
                 // read the file
                 //model.read( ontURL );
                 // print validation report
                 //ValidityReport report = model.validate();
                 //printIterator( report.getReports(), "Validation Results" );
                 System.out.println("Back");

                 // print superclasses using the utility function
                 //OntClass c = model.getOntClass( ontURL + "#Building" );
                 //printIterator(c.listSuperClasses(), "All super classes of " +
c.getLocalName());
                 // OntClass provides function to print *only* the direct subclasses
                 //printIterator(c.listSuperClasses(true), "Direct superclasses of " +
c.getLocalName());

                 System.out.println();
//pellet
                return inf;
        }

        public static String testInfer(InfModel inf, Resource subj, Property pred,
Resource obj) {

                String testCondition;
                if(inf.contains(subj, pred, obj)){
                        testCondition = "true";
                }
                else{
                        testCondition = "false";
                }
                return(testCondition);
        }

        public static Model createResultModel(Model m, Resource s, Property p,
Resource o){
                Model tempMod = ModelFactory.createDefaultModel();
                for (StmtIterator i = m.listStatements(s,p,o); i.hasNext();){
                        Statement stmt = i.nextStatement();
                        //String str = stmt.getSubject().toString();
                        //System.out.println(str);
                        tempMod.add(stmt);
                }
                System.out.println("CreateTempResultModel OK");

                //N-Triple writer
//              System.out.println("*-* n-triple writer *-*");
```

```
//              tempMod.write(System.out, "N-TRIPLE");
                return tempMod;
        }

        public static ResultSet runSimpleSparql(Model m, String subCond, String
predCond, String objCond){
                System.out.println("*** Executing SPARQL ***");
                //Run SPARQL to retrieve all subClassOf
                String sparql =
//                      "PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>"+
                        "PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>"+
//                      "PREFIX jkb2: <file:///java_ws/edconf/ed_map.n3#>"+
                        "PREFIX jkb: <http://localhost/edbuildings.owl#>"+
                        "PREFIX owl: <http://www.w3.org/2002/07/owl#>"+
                        "SELECT DISTINCT ?s WHERE {" +
///
//                      "?s rdf:type ?o ."+
//                      "FILTER regex(?s, \"bldno\", \"i\")" +
///

                        "       " + subCond +" "+ predCond +" " + objCond +  " ." +
                        "}";

                Query q = QueryFactory.create(sparql);
                ResultSet rs = QueryExecutionFactory.create(q, m).execSelect();
                return rs;
        }

        public static List parseSparqlForGID(ResultSet rs){

                int count = 0;
                //Create a list to take the subjects
                List l = new LinkedList(); //A Doubly-linked list
                l = new ArrayList(); //List implemented as a growable array

                while (rs.hasNext()){
                        QuerySolution row = rs.nextSolution();
                        //System.out.println(row.toString());

                        //Split SPARQL result rows extracting all the subject resources
                        int int1 = row.toString().indexOf("bldno");
                        int int2 = row.toString().indexOf(">");
                        String split = row.toString().substring(int1+5, int2);
                        l.add(count,split);
                        count++;
                }

                System.out.println("List size : "+ l.size());
                for (int k = l.size()-1; k>=0; k--){
                        System.out.println(l.get(k).toString());
                }
                System.out.println("RunSimpleSparql OK");
        return l;
        }

        public static Connection dbConnect(String url, String uname, String pwd)
throws SQLException{
                Connection dbConn = DriverManager.getConnection(url, uname, pwd);
                return dbConn;
        }

        public static Vector queryDB(List list, Connection db, double buffer, MyPoint
pt) throws SQLException{
                //Returns a vector filled with centroid xCoord, centroid yCoord, gid
                String query;
                String columns = "gid";
                String table = "edinburgh";
                Double x = pt.getX();
                Double y = pt.getY();
                String whereClause = listToString(list, ",");
                whereClause = " AND " + columns + " in (" + whereClause+")";
                        query = "SELECT astext (st_centroid(the_geom))," + columns + "
FROM " + table + " WHERE " +
                        "the_geom && Expand(GeomFromText('POINT(" + x + " " + y +")',
27700),"+buffer + ") AND " +
                        "Distance(GeomFromText('POINT(" + x + " " + y +")', 27700),
```

```
the_geom)<"+buffer + whereClause;

                System.out.println(query);
                java.sql.Statement st = db.createStatement();
                java.sql.ResultSet rs2 = st.executeQuery(query);
                Vector vRS = new Vector();
                int index1, index2, index3;
                String xCoord, yCoord;
                System.out.println("queryDB ResultSet");
                while (rs2.next()){
                        //Add result set to a vector as the RS does not persist
                        // process centroid column (1)
                        System.out.println("DB Result set " + rs2.getString(1));
                        String point = rs2.getString(1);
                        index1 = point.indexOf('(');
                        index2 = point.indexOf(' ');
                        index3 = point.indexOf(')');
                        xCoord = point.substring(index1+1, index2);
                        //xCoord = "325555.00";
                        vRS.add(xCoord);
                        yCoord = point.substring(index2+1, index3);
                        vRS.add(yCoord);
                        //MyPoint centroid = new MyPoint(xCoord, yCoord);
                        //vRS.add(centroid);
                        //result set index starts at 2 - gid
                        vRS.add(rs2.getString(2));
                }
                return vRS;
        }

        public static String listToString(List l, String separator) {
                // Convert an array of strings to one string.
                // Put the 'separator' string between each element.
                //enclose each token in a single quote for query purposes

           StringBuffer result = new StringBuffer();
           if (l.size() > 0) {
               result.append("'"+list.get(0).toString()+"'");
               for (int i=1; i<l.size(); i++) {
                   result.append(separator);
                   result.append("'"+l.get(i).toString()+"'");
               }
           }
           return result.toString();
        }

        public static void printStatements(Model m, Resource s, Property p, Resource
o){
                int count = 1;
                for (StmtIterator i = m.listStatements(s,p,o);i.hasNext();){
                        Statement stmt = i.nextStatement();
                        System.out.println(" - " + PrintUtil.print(stmt));
//                      System.out.println("Statement count: " + count);
                        count++;
                }
                System.out.println("Statement count: " + count);
        }

    public static void printIterator(Iterator i, String header) {
        System.out.println(header);
        for(int c = 0; c < header.length(); c++)
            System.out.print("=");
        System.out.println();

        if(i.hasNext()) {
                while (i.hasNext())
                    System.out.println( i.next() );
        }
        else
            System.out.println("<EMPTY>");
        System.out.println();
    }
}
```

EdSchema.java

```
package uk.ac.ed;
/* CVS $Id: $ */

import com.hp.hpl.jena.rdf.model.*;
import com.hp.hpl.jena.ontology.*;
/**
 * Vocabulary definitions from edbuildontology.owl
 * @author Auto-generated by schemagen on 19 Nov 2007 13:09
 */
public class EdSchema {
    /** <p>The ontology model that holds the vocabulary terms</p> */
    private static OntModel m_model = ModelFactory.createOntologyModel(
OntModelSpec.OWL_MEM, null );

    /** <p>The namespace of the vocabulary as a string</p> */
    public static final String NS = "http://eeo.ed.ac.uk/jkb/";

    /** <p>The namespace of the vocabulary as a string</p>
     *  @see #NS */
    public static String getURI() {return NS;}

    /** <p>The namespace of the vocabulary as a resource</p> */
    public static final Resource NAMESPACE = m_model.createResource( NS );

    public static final ObjectProperty isBusinessTypeOf =
m_model.createObjectProperty(
"http://eeo.ed.ac.uk/jkb/edbuildontology.owl#isBusinessTypeOf" );

    public static final ObjectProperty hasBuildingName =
m_model.createObjectProperty(
"http://eeo.ed.ac.uk/jkb/edbuildontology.owl#hasBuildingName" );

    public static final ObjectProperty hasFoodService = m_model.createObjectProperty(
"http://eeo.ed.ac.uk/jkb/edbuildontology.owl#hasFoodService" );

    public static final ObjectProperty hasAlcoholLicense =
m_model.createObjectProperty(
"http://eeo.ed.ac.uk/jkb/edbuildontology.owl#hasAlcoholLicense" );

    public static final ObjectProperty hasBuildingFunction =
m_model.createObjectProperty(
"http://eeo.ed.ac.uk/jkb/edbuildontology.owl#hasBuildingFunction" );

    public static final ObjectProperty isBuildingUnitOf =
m_model.createObjectProperty(
"http://eeo.ed.ac.uk/jkb/edbuildontology.owl#isBuildingUnitOf" );

    public static final ObjectProperty isBuildingFunctionOf =
m_model.createObjectProperty(
"http://eeo.ed.ac.uk/jkb/edbuildontology.owl#isBuildingFunctionOf" );

    public static final ObjectProperty hasBusinessType =
m_model.createObjectProperty(
"http://eeo.ed.ac.uk/jkb/edbuildontology.owl#hasBusinessType" );

    public static final ObjectProperty hasGroceries = m_model.createObjectProperty(
"http://eeo.ed.ac.uk/jkb/edbuildontology.owl#hasGroceries" );

    public static final ObjectProperty hasBuildingUnit =
m_model.createObjectProperty(
"http://eeo.ed.ac.uk/jkb/edbuildontology.owl#hasBuildingUnit" );

    public static final DatatypeProperty hasFloospace =
m_model.createDatatypeProperty(
"http://eeo.ed.ac.uk/jkb/edbuildontology.owl#hasFloospace" );

    public static final DatatypeProperty hasFootprint =
m_model.createDatatypeProperty(
"http://eeo.ed.ac.uk/jkb/edbuildontology.owl#hasFootprint" );

    public static final DatatypeProperty hasBuildDate =
m_model.createDatatypeProperty(
```

```
"http://eeo.ed.ac.uk/jkb/edbuildontology.owl#hasBuildDate" );

    public static final DatatypeProperty hasHeight = m_model.createDatatypeProperty(
"http://eeo.ed.ac.uk/jkb/edbuildontology.owl#hasHeight" );

    public static final OntClass Recreation = m_model.createClass(
"http://eeo.ed.ac.uk/jkb/edbuildontology.owl#Recreation" );

    public static final OntClass Grocer = m_model.createClass(
"http://eeo.ed.ac.uk/jkb/edbuildontology.owl#Grocer" );

    public static final OntClass Retail = m_model.createClass(
"http://eeo.ed.ac.uk/jkb/edbuildontology.owl#Retail" );

    public static final OntClass Restaurant = m_model.createClass(
"http://eeo.ed.ac.uk/jkb/edbuildontology.owl#Restaurant" );

    public static final OntClass Hostel = m_model.createClass(
"http://eeo.ed.ac.uk/jkb/edbuildontology.owl#Hostel" );

    public static final OntClass ClothesStore = m_model.createClass(
"http://eeo.ed.ac.uk/jkb/edbuildontology.owl#ClothesStore" );

    public static final OntClass BookStore = m_model.createClass(
"http://eeo.ed.ac.uk/jkb/edbuildontology.owl#BookStore" );

    public static final OntClass Newsagent = m_model.createClass(
"http://eeo.ed.ac.uk/jkb/edbuildontology.owl#Newsagent" );

    public static final OntClass Omnivorous = m_model.createClass(
"http://eeo.ed.ac.uk/jkb/edbuildontology.owl#Omnivorous" );

    public static final OntClass Theatre = m_model.createClass(
"http://eeo.ed.ac.uk/jkb/edbuildontology.owl#Theatre" );

    public static final OntClass ConcertVenue = m_model.createClass(
"http://eeo.ed.ac.uk/jkb/edbuildontology.owl#ConcertVenue" );

    public static final OntClass Bakery = m_model.createClass(
"http://eeo.ed.ac.uk/jkb/edbuildontology.owl#Bakery" );

    public static final OntClass Eatery = m_model.createClass(
"http://eeo.ed.ac.uk/jkb/edbuildontology.owl#Eatery" );

    public static final OntClass Financial = m_model.createClass(
"http://eeo.ed.ac.uk/jkb/edbuildontology.owl#Financial" );

    public static final OntClass NightClub = m_model.createClass(
"http://eeo.ed.ac.uk/jkb/edbuildontology.owl#NightClub" );

    public static final OntClass Bar = m_model.createClass(
"http://eeo.ed.ac.uk/jkb/edbuildontology.owl#Bar" );

    public static final OntClass CoffeeShop = m_model.createClass(
"http://eeo.ed.ac.uk/jkb/edbuildontology.owl#CoffeeShop" );

    public static final OntClass Cinema = m_model.createClass(
"http://eeo.ed.ac.uk/jkb/edbuildontology.owl#Cinema" );

    public static final OntClass BureauDeChange = m_model.createClass(
"http://eeo.ed.ac.uk/jkb/edbuildontology.owl#BureauDeChange" );

    public static final OntClass Private = m_model.createClass(
"http://eeo.ed.ac.uk/jkb/edbuildontology.owl#Private" );

    public static final OntClass ConvenienceStore = m_model.createClass(
"http://eeo.ed.ac.uk/jkb/edbuildontology.owl#ConvenienceStore" );

    public static final OntClass Commercial = m_model.createClass(
"http://eeo.ed.ac.uk/jkb/edbuildontology.owl#Commercial" );

    public static final OntClass BuildingSociety = m_model.createClass(
"http://eeo.ed.ac.uk/jkb/edbuildontology.owl#BuildingSociety" );

    public static final OntClass LicensedPremises = m_model.createClass(
"http://eeo.ed.ac.uk/jkb/edbuildontology.owl#LicensedPremises" );
```

```
    public static final OntClass Public = m_model.createClass(
"http://eeo.ed.ac.uk/jkb/edbuildontology.owl#Public" );

    public static final OntClass Bistro = m_model.createClass(
"http://eeo.ed.ac.uk/jkb/edbuildontology.owl#Bistro" );

    public static final OntClass Butcher = m_model.createClass(
"http://eeo.ed.ac.uk/jkb/edbuildontology.owl#Butcher" );

    public static final OntClass BedAndBreakfast = m_model.createClass(
"http://eeo.ed.ac.uk/jkb/edbuildontology.owl#BedAndBreakfast" );

    public static final OntClass Motel = m_model.createClass(
"http://eeo.ed.ac.uk/jkb/edbuildontology.owl#Motel" );

    public static final OntClass Bank = m_model.createClass(
"http://eeo.ed.ac.uk/jkb/edbuildontology.owl#Bank" );

    public static final OntClass BuildingUnit = m_model.createClass(
"http://eeo.ed.ac.uk/jkb/edbuildontology.owl#BuildingUnit" );

    public static final OntClass Building = m_model.createClass(
"http://eeo.ed.ac.uk/jkb/edbuildontology.owl#Building" );

    public static final OntClass WineBar = m_model.createClass(
"http://eeo.ed.ac.uk/jkb/edbuildontology.owl#WineBar" );

    public static final OntClass BuildingForm = m_model.createClass(
"http://eeo.ed.ac.uk/jkb/edbuildontology.owl#BuildingForm" );

    public static final OntClass Accommodation = m_model.createClass(
"http://eeo.ed.ac.uk/jkb/edbuildontology.owl#Accommodation" );

    public static final OntClass Cafe = m_model.createClass(
"http://eeo.ed.ac.uk/jkb/edbuildontology.owl#Cafe" );

    public static final OntClass BuildingUnitFunction = m_model.createClass(
"http://eeo.ed.ac.uk/jkb/edbuildontology.owl#BuildingUnitFunction" );

    public static final OntClass CocktailBar = m_model.createClass(
"http://eeo.ed.ac.uk/jkb/edbuildontology.owl#CocktailBar" );

    public static final OntClass Supermarket = m_model.createClass(
"http://eeo.ed.ac.uk/jkb/edbuildontology.owl#Supermarket" );

    public static final OntClass Hotel = m_model.createClass(
"http://eeo.ed.ac.uk/jkb/edbuildontology.owl#Hotel" );

    public static final OntClass ShoppingCentre = m_model.createClass(
"http://eeo.ed.ac.uk/jkb/edbuildontology.owl#ShoppingCentre" );

    public static final OntClass Vegetarian = m_model.createClass(
"http://eeo.ed.ac.uk/jkb/edbuildontology.owl#Vegetarian" );

}
```

Edbuildings.java

```
public static final String NS = "http://localhost/";

    /** <p>The namespace of the vocabulary as a string</p>
     *  @see #NS */
    public static String getURI() {return NS;}

    /** <p>The namespace of the vocabulary as a resource</p> */
    public static final Resource NAMESPACE = m_model.createResource( NS );

    public static final ObjectProperty hasUnit = m_model.createObjectProperty(
"http://localhost/edbuildings.owl#hasUnit" );

    public static final DatatypeProperty hasBusinessType =
m_model.createDatatypeProperty( "http://localhost/edbuildings.owl#hasBusinessType" );
```

```
    public static final ObjectProperty hasAlcoholLicense =
m_model.createObjectProperty( "http://localhost/edbuildings.owl#hasAlcoholLicense" );

    public static final DatatypeProperty hasBusinessName =
m_model.createDatatypeProperty( "http://localhost/edbuildings.owl#hasBusinessName" );

    public static final DatatypeProperty hasCloseTime =
m_model.createDatatypeProperty( "http://localhost/edbuildings.owl#hasCloseTime" );

    public static final DatatypeProperty hasBuildingID =
m_model.createDatatypeProperty( "http://localhost/edbuildings.owl#hasBuildingID" );

    public static final DatatypeProperty hasConstructDate =
m_model.createDatatypeProperty( "http://localhost/edbuildings.owl#hasConstructDate"
);

    public static final DatatypeProperty isOccupied = m_model.createDatatypeProperty(
"http://localhost/edbuildings.owl#isOccupied" );

    public static final DatatypeProperty hasBuildingHeight =
m_model.createDatatypeProperty( "http://localhost/edbuildings.owl#hasBuildingHeight"
);

    public static final DatatypeProperty hasFloorspace =
m_model.createDatatypeProperty( "http://localhost/edbuildings.owl#hasFloorspace" );

    public static final DatatypeProperty hasFloorLevel =
m_model.createDatatypeProperty( "http://localhost/edbuildings.owl#hasFloorLevel" );

    public static final DatatypeProperty hasBusinessID =
m_model.createDatatypeProperty( "http://localhost/edbuildings.owl#hasBusinessID" );

    public static final DatatypeProperty hasOpenTime =
m_model.createDatatypeProperty( "http://localhost/edbuildings.owl#hasOpenTime" );

    public static final DatatypeProperty hasFoodService =
m_model.createDatatypeProperty( "http://localhost/edbuildings.owl#hasFoodService" );

    public static final DatatypeProperty hasStoreys = m_model.createDatatypeProperty(
"http://localhost/edbuildings.owl#hasStoreys" );

    public static final DatatypeProperty hasFootprint =
m_model.createDatatypeProperty( "http://localhost/edbuildings.owl#hasFootprint" );

    public static final DatatypeProperty hasOccupyDate =
m_model.createDatatypeProperty( "http://localhost/edbuildings.owl#hasOccupyDate" );

    public static final DatatypeProperty hasBuildingName =
m_model.createDatatypeProperty( "http://localhost/edbuildings.owl#hasBuildingName" );

    public static final OntClass Newsagent = m_model.createClass(
"http://localhost/edbuildings.owl#Newsagent" );

    public static final OntClass BureauDeChange = m_model.createClass(
"http://localhost/edbuildings.owl#BureauDeChange" );

    public static final OntClass Financial = m_model.createClass(
"http://localhost/edbuildings.owl#Financial" );

    public static final OntClass Bank = m_model.createClass(
"http://localhost/edbuildings.owl#Bank" );

    public static final OntClass BedAndBreakfast = m_model.createClass(
"http://localhost/edbuildings.owl#BedAndBreakfast" );

    public static final OntClass Butcher = m_model.createClass(
"http://localhost/edbuildings.owl#Butcher" );

    public static final OntClass ShoppingCentre = m_model.createClass(
"http://localhost/edbuildings.owl#ShoppingCentre" );

    public static final OntClass Cinema = m_model.createClass(
"http://localhost/edbuildings.owl#Cinema" );

    public static final OntClass Eatery = m_model.createClass(
```

```
"http://localhost/edbuildings.owl#Eatery" );

    public static final OntClass Bakery = m_model.createClass(
"http://localhost/edbuildings.owl#Bakery" );

    public static final OntClass Recreation = m_model.createClass(
"http://localhost/edbuildings.owl#Recreation" );

    public static final OntClass Grocer = m_model.createClass(
"http://localhost/edbuildings.owl#Grocer" );

    public static final OntClass BookStore = m_model.createClass(
"http://localhost/edbuildings.owl#BookStore" );

    public static final OntClass Hotel = m_model.createClass(
"http://localhost/edbuildings.owl#Hotel" );

    public static final OntClass BuildingSociety = m_model.createClass(
"http://localhost/edbuildings.owl#BuildingSociety" );

    public static final OntClass CoffeeShop = m_model.createClass(
"http://localhost/edbuildings.owl#CoffeeShop" );

    public static final OntClass ConvenienceStore = m_model.createClass(
"http://localhost/edbuildings.owl#ConvenienceStore" );

    public static final OntClass Bar = m_model.createClass(
"http://localhost/edbuildings.owl#Bar" );

    public static final OntClass VegetarianRestaurant = m_model.createClass(
"http://localhost/edbuildings.owl#VegetarianRestaurant" );

    public static final OntClass ConcertVenue = m_model.createClass(
"http://localhost/edbuildings.owl#ConcertVenue" );

    public static final OntClass WineBar = m_model.createClass(
"http://localhost/edbuildings.owl#WineBar" );

    public static final OntClass ClothesStore = m_model.createClass(
"http://localhost/edbuildings.owl#ClothesStore" );

    public static final OntClass Retail = m_model.createClass(
"http://localhost/edbuildings.owl#Retail" );

    public static final OntClass Motel = m_model.createClass(
"http://localhost/edbuildings.owl#Motel" );

    public static final OntClass Hostel = m_model.createClass(
"http://localhost/edbuildings.owl#Hostel" );

    public static final OntClass Commercial = m_model.createClass(
"http://localhost/edbuildings.owl#Commercial" );

    public static final OntClass Private = m_model.createClass(
"http://localhost/edbuildings.owl#Private" );

    public static final OntClass Accommodation = m_model.createClass(
"http://localhost/edbuildings.owl#Accommodation" );

    public static final OntClass Theatre = m_model.createClass(
"http://localhost/edbuildings.owl#Theatre" );

    public static final OntClass Bistro = m_model.createClass(
"http://localhost/edbuildings.owl#Bistro" );

    public static final OntClass Restaurant = m_model.createClass(
"http://localhost/edbuildings.owl#Restaurant" );

    public static final OntClass CocktailBar = m_model.createClass(
"http://localhost/edbuildings.owl#CocktailBar" );

    public static final OntClass Public = m_model.createClass(
"http://localhost/edbuildings.owl#Public" );

    public static final OntClass Cafe = m_model.createClass(
"http://localhost/edbuildings.owl#Cafe" );
```

```
    public static final OntClass Supermarket = m_model.createClass(
"http://localhost/edbuildings.owl#Supermarket" );

    public static final OntClass NightClub = m_model.createClass(
"http://localhost/edbuildings.owl#NightClub" );

    public static final OntClass Building = m_model.createClass(
"http://localhost/edbuildings.owl#Building" );

    public static final OntClass Omnivorous = m_model.createClass(
"http://localhost/edbuildings.owl#Omnivorous" );

    public static final OntClass PublicHouses = m_model.createClass(
"http://localhost/edbuildings.owl#PublicHouses" );

}
```

edbuildings.owl

```
<?xml version="1.0"?>
<rdf:RDF
    xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
    xmlns:xsd="http://www.w3.org/2001/XMLSchema#"
    xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
    xmlns:owl="http://www.w3.org/2002/07/owl#"
    xmlns="http://localhost/edbuildings.owl#"
  xml:base="http://localhost/edbuildings.owl">
  <owl:Ontology rdf:about=""/>
  <owl:Class rdf:ID="Building">
    <rdfs:subClassOf rdf:resource="http://www.w3.org/2002/07/owl#Thing"/>
    <rdfs:subClassOf>
      <owl:Restriction>
        <owl:someValuesFrom>
          <owl:Class>
            <owl:unionOf rdf:parseType="Collection">
              <owl:Class rdf:ID="Commercial"/>
              <owl:Class rdf:ID="Public"/>
              <owl:Class rdf:ID="Private"/>
            </owl:unionOf>
          </owl:Class>
        </owl:someValuesFrom>
        <owl:onProperty>
          <owl:TransitiveProperty rdf:ID="hasUnit"/>
        </owl:onProperty>
      </owl:Restriction>
    </rdfs:subClassOf>
  </owl:Class>
  <owl:Class rdf:ID="Hotel">
    <rdfs:subClassOf>
      <owl:Class rdf:ID="Accommodation"/>
    </rdfs:subClassOf>
  </owl:Class>
  <owl:Class rdf:ID="CoffeeShop">
    <rdfs:subClassOf>
      <owl:Class rdf:ID="Eatery"/>
    </rdfs:subClassOf>
  </owl:Class>
  <owl:Class rdf:ID="Omnivorous">
    <owl:equivalentClass>
      <owl:Class>
        <owl:complementOf>
          <owl:Class rdf:ID="VegetarianRestaurant"/>
        </owl:complementOf>
      </owl:Class>
    </owl:equivalentClass>
    <rdfs:subClassOf>
      <owl:Class rdf:ID="Restaurant"/>
    </rdfs:subClassOf>
  </owl:Class>
  <owl:Class rdf:ID="Bank">
    <rdfs:subClassOf>
      <owl:Class rdf:ID="Financial"/>
```

```
    </rdfs:subClassOf>
  </owl:Class>
  <owl:Class rdf:ID="Bistro">
    <rdfs:subClassOf>
      <owl:Class rdf:about="#Eatery"/>
    </rdfs:subClassOf>
  </owl:Class>
  <owl:Class rdf:ID="Cafe">
    <rdfs:subClassOf>
      <owl:Class rdf:about="#Eatery"/>
    </rdfs:subClassOf>
  </owl:Class>
  <owl:Class rdf:about="#Public">
    <owl:disjointWith>
      <owl:Class rdf:about="#Commercial"/>
    </owl:disjointWith>
    <owl:disjointWith>
      <owl:Class rdf:about="#Private"/>
    </owl:disjointWith>
    <rdfs:subClassOf rdf:resource="#Building"/>
  </owl:Class>
  <owl:Class rdf:ID="CocktailBar">
    <rdfs:subClassOf>
      <owl:Class rdf:ID="PublicHouses"/>
    </rdfs:subClassOf>
  </owl:Class>
  <owl:Class rdf:about="#VegetarianRestaurant">
    <rdfs:subClassOf>
      <owl:Class rdf:about="#Restaurant"/>
    </rdfs:subClassOf>
  </owl:Class>
  <owl:Class rdf:about="#Accommodation">
    <owl:disjointWith>
      <owl:Class rdf:ID="Recreation"/>
    </owl:disjointWith>
    <owl:disjointWith>
      <owl:Class rdf:ID="Retail"/>
    </owl:disjointWith>
    <owl:disjointWith>
      <owl:Class rdf:about="#PublicHouses"/>
    </owl:disjointWith>
    <owl:disjointWith>
      <owl:Class rdf:about="#Financial"/>
    </owl:disjointWith>
    <rdfs:subClassOf>
      <owl:Class rdf:about="#Commercial"/>
    </rdfs:subClassOf>
    <owl:disjointWith>
      <owl:Class rdf:about="#Eatery"/>
    </owl:disjointWith>
  </owl:Class>
  <owl:Class rdf:ID="ClothesStore">
    <rdfs:subClassOf>
      <owl:Class rdf:about="#Retail"/>
    </rdfs:subClassOf>
  </owl:Class>
  <owl:Class rdf:about="#Eatery">
    <rdfs:subClassOf>
      <owl:Restriction>
        <owl:onProperty>
          <owl:ObjectProperty rdf:ID="hasFoodService"/>
        </owl:onProperty>
        <owl:hasValue rdf:datatype="http://www.w3.org/2001/XMLSchema#boolean"
        >true</owl:hasValue>
      </owl:Restriction>
    </rdfs:subClassOf>
    <owl:disjointWith>
      <owl:Class rdf:about="#Retail"/>
    </owl:disjointWith>
    <owl:disjointWith>
      <owl:Class rdf:about="#PublicHouses"/>
    </owl:disjointWith>
    <owl:disjointWith>
      <owl:Class rdf:about="#Financial"/>
    </owl:disjointWith>
    <rdfs:subClassOf>
```

```
      <owl:Class rdf:about="#Commercial"/>
    </rdfs:subClassOf>
    <owl:disjointWith>
      <owl:Class rdf:about="#Recreation"/>
    </owl:disjointWith>
    <owl:disjointWith rdf:resource="#Accommodation"/>
  </owl:Class>
  <owl:Class rdf:ID="Hostel">
    <rdfs:subClassOf rdf:resource="#Accommodation"/>
  </owl:Class>
  <owl:Class rdf:ID="BureauDeChange">
    <rdfs:subClassOf>
      <owl:Class rdf:about="#Financial"/>
    </rdfs:subClassOf>
  </owl:Class>
  <owl:Class rdf:ID="Grocer">
    <rdfs:subClassOf>
      <owl:Class rdf:about="#Retail"/>
    </rdfs:subClassOf>
  </owl:Class>
  <owl:Class rdf:ID="ConcertVenue">
    <rdfs:subClassOf>
      <owl:Class rdf:about="#Recreation"/>
    </rdfs:subClassOf>
  </owl:Class>
  <owl:Class rdf:ID="ConvenienceStore">
    <rdfs:subClassOf>
      <owl:Class rdf:about="#Retail"/>
    </rdfs:subClassOf>
  </owl:Class>
  <owl:Class rdf:ID="Newsagent">
    <rdfs:subClassOf>
      <owl:Class rdf:about="#Retail"/>
    </rdfs:subClassOf>
  </owl:Class>
  <owl:Class rdf:ID="Bakery">
    <rdfs:subClassOf>
      <owl:Class rdf:about="#Retail"/>
    </rdfs:subClassOf>
  </owl:Class>
  <owl:Class rdf:about="#Recreation">
    <owl:disjointWith>
      <owl:Class rdf:about="#Financial"/>
    </owl:disjointWith>
    <owl:disjointWith rdf:resource="#Eatery"/>
    <owl:disjointWith>
      <owl:Class rdf:about="#PublicHouses"/>
    </owl:disjointWith>
    <rdfs:subClassOf>
      <owl:Class rdf:about="#Commercial"/>
    </rdfs:subClassOf>
    <owl:disjointWith rdf:resource="#Accommodation"/>
    <owl:disjointWith>
      <owl:Class rdf:about="#Retail"/>
    </owl:disjointWith>
  </owl:Class>
  <owl:Class rdf:about="#Commercial">
    <owl:disjointWith>
      <owl:Class rdf:about="#Private"/>
    </owl:disjointWith>
    <owl:disjointWith rdf:resource="#Public"/>
    <rdfs:subClassOf rdf:resource="#Building"/>
  </owl:Class>
  <owl:Class rdf:ID="BuildingSociety">
    <rdfs:subClassOf>
      <owl:Class rdf:about="#Financial"/>
    </rdfs:subClassOf>
  </owl:Class>
  <owl:Class rdf:ID="ShoppingCentre">
    <rdfs:subClassOf>
      <owl:Class rdf:about="#Retail"/>
    </rdfs:subClassOf>
  </owl:Class>
  <owl:Class rdf:ID="Motel">
    <rdfs:subClassOf rdf:resource="#Accommodation"/>
  </owl:Class>
```

```
<owl:Class rdf:about="#Financial">
  <owl:disjointWith>
    <owl:Class rdf:about="#PublicHouses"/>
  </owl:disjointWith>
  <owl:disjointWith>
    <owl:Class rdf:about="#Retail"/>
  </owl:disjointWith>
  <owl:disjointWith rdf:resource="#Eatery"/>
  <owl:disjointWith rdf:resource="#Recreation"/>
  <rdfs:subClassOf rdf:resource="#Commercial"/>
  <owl:disjointWith rdf:resource="#Accommodation"/>
</owl:Class>
<owl:Class rdf:about="#Retail">
  <owl:disjointWith rdf:resource="#Eatery"/>
  <owl:disjointWith rdf:resource="#Accommodation"/>
  <owl:disjointWith rdf:resource="#Recreation"/>
  <rdfs:subClassOf rdf:resource="#Commercial"/>
  <owl:disjointWith rdf:resource="#Financial"/>
  <owl:disjointWith>
    <owl:Class rdf:about="#PublicHouses"/>
  </owl:disjointWith>
</owl:Class>
<owl:Class rdf:ID="Cinema">
  <rdfs:subClassOf rdf:resource="#Recreation"/>
</owl:Class>
<owl:Class rdf:ID="NightClub">
  <rdfs:subClassOf rdf:resource="#Recreation"/>
</owl:Class>
<owl:Class rdf:ID="Bar">
  <owl:sameAs rdf:resource="http://localhost/edbuildings.owl#Pub"/>
  <rdfs:subClassOf>
    <owl:Class rdf:about="#PublicHouses"/>
  </rdfs:subClassOf>
</owl:Class>
<owl:Class rdf:ID="BedAndBreakfast">
  <rdfs:subClassOf rdf:resource="#Accommodation"/>
</owl:Class>
<owl:Class rdf:ID="Theatre">
  <rdfs:subClassOf rdf:resource="#Recreation"/>
</owl:Class>
<owl:Class rdf:about="#Restaurant">
  <rdfs:subClassOf rdf:resource="#Eatery"/>
</owl:Class>
<owl:Class rdf:ID="Butcher">
  <rdfs:subClassOf rdf:resource="#Retail"/>
</owl:Class>
<owl:Class rdf:ID="WineBar">
  <rdfs:subClassOf>
    <owl:Class rdf:about="#PublicHouses"/>
  </rdfs:subClassOf>
</owl:Class>
<owl:Class rdf:ID="BookStore">
  <rdfs:subClassOf rdf:resource="#Retail"/>
</owl:Class>
<owl:Class rdf:ID="Supermarket">
  <rdfs:subClassOf rdf:resource="#Retail"/>
</owl:Class>
<owl:Class rdf:about="#PublicHouses">
  <owl:disjointWith rdf:resource="#Accommodation"/>
  <owl:disjointWith rdf:resource="#Eatery"/>
  <owl:disjointWith rdf:resource="#Retail"/>
  <owl:disjointWith rdf:resource="#Financial"/>
  <owl:disjointWith rdf:resource="#Recreation"/>
  <rdfs:subClassOf rdf:resource="#Commercial"/>
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:hasValue rdf:datatype="http://www.w3.org/2001/XMLSchema#boolean"
      >true</owl:hasValue>
      <owl:onProperty>
        <owl:DatatypeProperty rdf:ID="hasAlcoholLicense"/>
      </owl:onProperty>
    </owl:Restriction>
  </rdfs:subClassOf>
</owl:Class>
<owl:Class rdf:about="#Private">
  <rdfs:subClassOf rdf:resource="#Building"/>
```

```
    <owl:disjointWith rdf:resource="#Public"/>
    <owl:disjointWith rdf:resource="#Commercial"/>
  </owl:Class>
  <owl:ObjectProperty rdf:about="#hasFoodService">
    <rdfs:domain rdf:resource="#Eatery"/>
  </owl:ObjectProperty>
  <owl:DatatypeProperty rdf:ID="hasFloorLevel">
    <rdfs:domain rdf:resource="#Commercial"/>
  </owl:DatatypeProperty>
  <owl:DatatypeProperty rdf:ID="isOccupied">
    <rdfs:domain rdf:resource="#Commercial"/>
  </owl:DatatypeProperty>
  <owl:DatatypeProperty rdf:ID="hasOpenTime">
    <rdfs:domain rdf:resource="#Commercial"/>
  </owl:DatatypeProperty>
  <owl:DatatypeProperty rdf:ID="hasCloseTime">
    <rdfs:domain rdf:resource="#Commercial"/>
  </owl:DatatypeProperty>
  <owl:DatatypeProperty rdf:ID="hasBusinessType">
    <rdfs:domain rdf:resource="#Commercial"/>
  </owl:DatatypeProperty>
  <owl:DatatypeProperty rdf:ID="hasConstructDate">
    <rdfs:domain rdf:resource="#Building"/>
  </owl:DatatypeProperty>
  <owl:DatatypeProperty rdf:ID="hasBuildingName">
    <rdfs:domain rdf:resource="#Building"/>
  </owl:DatatypeProperty>
  <owl:DatatypeProperty rdf:ID="hasBuildingID">
    <rdfs:domain rdf:resource="#Building"/>
  </owl:DatatypeProperty>
  <owl:DatatypeProperty rdf:ID="hasBuildingHeight">
    <rdfs:domain rdf:resource="#Building"/>
  </owl:DatatypeProperty>
  <owl:DatatypeProperty rdf:ID="hasBusinessName">
    <rdfs:domain rdf:resource="#Commercial"/>
  </owl:DatatypeProperty>
  <owl:DatatypeProperty rdf:ID="hasStoreys">
    <rdfs:domain rdf:resource="#Building"/>
  </owl:DatatypeProperty>
  <owl:DatatypeProperty rdf:ID="hasOccupyDate">
    <rdfs:domain rdf:resource="#Commercial"/>
  </owl:DatatypeProperty>
  <owl:DatatypeProperty rdf:about="#hasAlcoholLicense">
    <rdfs:domain rdf:resource="#PublicHouses"/>
    <rdfs:range>
      <owl:Class>
        <owl:unionOf rdf:parseType="Collection">
          <owl:Class rdf:about="#Bar"/>
          <owl:Class rdf:about="#CocktailBar"/>
          <owl:Class rdf:about="#WineBar"/>
        </owl:unionOf>
      </owl:Class>
    </rdfs:range>
  </owl:DatatypeProperty>
  <owl:DatatypeProperty rdf:ID="hasBusinessID">
    <rdfs:domain rdf:resource="#Commercial"/>
  </owl:DatatypeProperty>
  <owl:DatatypeProperty rdf:ID="hasFloorspace">
    <rdfs:domain>
      <owl:Class>
        <owl:unionOf rdf:parseType="Collection">
          <owl:Class rdf:about="#Commercial"/>
          <owl:Class rdf:about="#Public"/>
          <owl:Class rdf:about="#Private"/>
        </owl:unionOf>
      </owl:Class>
    </rdfs:domain>
  </owl:DatatypeProperty>
  <owl:DatatypeProperty rdf:ID="hasFootprint">
    <rdfs:domain rdf:resource="#Building"/>
  </owl:DatatypeProperty>
  <owl:TransitiveProperty rdf:about="#hasUnit">
    <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#ObjectProperty"/>
    <rdfs:domain rdf:resource="#Building"/>
    <rdfs:range rdf:resource="#Building"/>
  </owl:TransitiveProperty>
```

```
</rdf:RDF>

<!-- Created with Protege (with OWL Plugin 2.1, Build 284)
http://protege.stanford.edu -->
```

ed_map.n3

```
@prefix map: <http://localhost/ed_map.n3#> .
@prefix db: <http://localhost/> .
@prefix vocab: <vocab/> .
@prefix jkb: <http://localhost/edbuildings.owl#> .
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
@prefix xsd: <http://www.w3.org/2001/XMLSchema#> .
@prefix d2rq: <http://www.wiwiss.fu-berlin.de/suhl/bizer/D2RQ/0.1#> .

map:database a d2rq:Database;
	d2rq:jdbcDriver "org.postgresql.Driver";
	d2rq:jdbcDSN "jdbc:postgresql://localhost:5432/pg_db_sp";
	d2rq:username "pguser";
	d2rq:password "pgpwd";
	.

# Table edinburgh
map:edinburgh a d2rq:ClassMap;
	d2rq:dataStorage map:database;
#	d2rq:uriPattern "@@edinburgh.gid@@";
    d2rq:uriPattern "http://eeo.ed.ac.uk/bldno@@edinburgh.gid@@";
	d2rq:class jkb:Building;
	.
map:edinburgh__label a d2rq:PropertyBridge;
	d2rq:belongsToClassMap map:edinburgh;
	d2rq:property rdfs:label;
	d2rq:pattern "building #@@edinburgh.gid@@";
	.
#map:edinburgh_gid a d2rq:PropertyBridge;
#	d2rq:belongsToClassMap map:edinburgh;
#	d2rq:property jkb:hasBuildingID;
#	d2rq:column "edinburgh.gid";
#	d2rq:datatype xsd:int;
#	.
#map:edinburgh_toid a d2rq:PropertyBridge;
#	d2rq:belongsToClassMap map:edinburgh;
#	d2rq:property vocab:edinburgh_toid;
#	d2rq:column "edinburgh.toid";
#	.
map:BusinessClassMap a d2rq:ClassMap;
    d2rq:uriPattern "bldno@@edinburgh.gid@@";
    d2rq:dataStorage map:database;
    .
map:BusinessType a d2rq:PropertyBridge;
    d2rq:property rdf:type;
    d2rq:uriPattern "http://localhost/edbuildings.owl#@@edinburgh.str_use_id@@";
    d2rq:belongsToClassMap map:BusinessClassMap;
    .
map:edinburgh_structname a d2rq:PropertyBridge;
	d2rq:belongsToClassMap map:edinburgh;
	d2rq:property jkb:hasBuidingName;
	d2rq:column "edinburgh.structname";
	.
map:edinburgh_areafloor a d2rq:PropertyBridge;
	d2rq:belongsToClassMap map:edinburgh;
	d2rq:property jkb:hasFloorspace;
	d2rq:column "edinburgh.areafloor";
	d2rq:datatype xsd:short;
	.
map:edinburgh_structhght a d2rq:PropertyBridge;
	d2rq:belongsToClassMap map:edinburgh;
	d2rq:property jkb:hasBuildingHeight;
	d2rq:column "edinburgh.structhght";
	d2rq:datatype xsd:short;
	.
map:edinburgh_no_levels a d2rq:PropertyBridge;
```

```
        d2rq:belongsToClassMap map:edinburgh;
        d2rq:property jkb:hasStoreys;
        d2rq:column "edinburgh.no_levels";
        d2rq:datatype xsd:short;
        .
map:edinburgh_built_date a d2rq:PropertyBridge;
        d2rq:belongsToClassMap map:edinburgh;
        d2rq:property jkb:hasConstructDate;
        d2rq:column "edinburgh.built_date";
        d2rq:datatype xsd:date;
        .
map:edinburgh_occup_date a d2rq:PropertyBridge;
        d2rq:belongsToClassMap map:edinburgh;
        d2rq:property jkb:hasOccupyDate;
        d2rq:column "edinburgh.occup_date";
        d2rq:datatype xsd:date;
        .
map:edinburgh_shape_area a d2rq:PropertyBridge;
        d2rq:belongsToClassMap map:edinburgh;
        d2rq:property jkb:hasFootprint;
        d2rq:column "edinburgh.shape_area";
        d2rq:datatype xsd:decimal;
        .
map:edinburgh_build_id a d2rq:PropertyBridge;
        d2rq:belongsToClassMap map:edinburgh;
        d2rq:property jkb:hasBuildingID;
        d2rq:column "edinburgh.build_id";
        d2rq:datatype xsd:short;
        .
#map:edinburgh_str_use_id a d2rq:PropertyBridge;
#       d2rq:belongsToClassMap map:edinburgh;
#       d2rq:property jkb:hasBusinessType;
#       d2rq:column "edinburgh.str_use_id";
#       .
map:edinburgh_licensed a d2rq:PropertyBridge;
        d2rq:belongsToClassMap map:edinburgh;
        d2rq:property jkb:hasAlcoholLicense;
        d2rq:column "edinburgh.licensed";
        .
map:edinburgh_bus_name a d2rq:PropertyBridge;
        d2rq:belongsToClassMap map:edinburgh;
        d2rq:property jkb:hasBusinessName;
        d2rq:column "edinburgh.bus_name";
        .
map:edinburgh_business_id a d2rq:PropertyBridge;
        d2rq:belongsToClassMap map:edinburgh;
        d2rq:property jkb:hasBusinessID;
        d2rq:column "edinburgh.business_id";
        d2rq:datatype xsd:short;
        .
map:edinburgh_open_time a d2rq:PropertyBridge;
        d2rq:belongsToClassMap map:edinburgh;
        d2rq:property jkb:hasOpenTime;
        d2rq:column "edinburgh.open_time";
        d2rq:datatype xsd:short;
        .
map:edinburgh_close_time a d2rq:PropertyBridge;
        d2rq:belongsToClassMap map:edinburgh;
        d2rq:property jkb:hasCloseTime;
        d2rq:column "edinburgh.close_time";
        d2rq:datatype xsd:short;
        .
#map:edinburgh_the_geom a d2rq:PropertyBridge;
#       d2rq:belongsToClassMap map:edinburgh;
#       d2rq:property vocab:edinburgh_the_geom;
#       d2rq:column "edinburgh.the_geom";
#       .
```

Appendix F – Subsequent research

A development of the Geographical Information Projects Index in line with the Scottish Geographic Information Strategy: "One Scotland One Geography."

David Bruce

MSc Thesis, University of Edinburgh

2008

In 2006, a partnership between The University of Edinburgh, The Scottish Government and The Association of Geographic Information in Scotland lead to the production of an online service designed to encourage diffusion of knowledge across the Scottish GI community and beyond. The project, part of the "One Scotland One Geography" initiative, aimed to create a service whereby metadata about projects could be shared in order to create synergies and reduce repetition across the GI community. The purpose of this project was to review and enhance a prototype Geographical Information Projects Registry (GIPR) to create a self-sustaining user-driven service. The resource exists as a database driven website which is accessed via the World Wide Web (WWW). The service can be found at www.gisprojects.net. The project aims to provide a more intelligent, user-driven website which requires little or no administration. The project aims to achieve this through highlighting synergies, using new methods to keep the database current and by providing alternative techniques of keeping users informed of new information obtained by the registry.

Adding Semantics to Geographic Data Models

Femke Reitsma
The e-Science Institute Public Lecture
In Association with eSI Thematic Programme: Spatial Semantics for Automating Geographic Information Processes, 07 April, 2009
e-Science Institute, 15 South College Street
Edinburgh

Meaning in geographic data sets can be found in three places: in the data set as a whole (as described by metadata), in the relationships among features in the data set (spatial or other), or in those features themselves. Most work on expressing geospatial semantics has focussed on the first two, this presentation is about the semantics to be found in individual features in a spatial data set.

Our traditional geospatial data models that allow us to represent individual features involve associating some measurable quality, such as temperature, or observable feature, such as a tree, with a point or region in space and time. When capturing data we implicitly subscribe to some kind of conceptualisation. If we can make this explicit in an ontology and associate it with the captured data, we can leverage formal semantics to reason with the concepts represented in our spatial data sets. This talk will present work on expressing semantics at the data model level and using these for discovering geospatial information, with a practical implementation as proof of concept.

Geophysical Research Abstracts,
Vol. 11, EGU2009-500, 2009
EGU General Assembly 2009

Extending Primitive Spatial Data Models to Include Semantics

F. Reitsma (1) and J Batcheller (2)

(1) University of Canterbury, Christchurch, New Zealand, (2) University of Edinburgh, Edinburgh, Scotland

Our traditional geospatial data model involves associating some measurable quality, such as temperature, or observable feature, such as a tree, with a point or region in space and time. When capturing data we implicitly subscribe to some kind of conceptualisation. If we can make this explicit in an ontology and associate it with the captured data, we can leverage formal semantics to reason with the concepts represented in our spatial data sets. To do so, we extend our fundamental representation of geospatial data in a data model by including a URI in our basic data model that links it to our ontology defining our conceptualisation, We thus extend Goodchild et al's geo-atom [1] with the addition of a URI: (x, Z, z(x), URI) . This provides us with pixel or feature level knowledge and the ability to create layers of data from a set of pixels or features that might be drawn from a database based on their semantics. Using open source tools, we present a prototype that involves simple reasoning as a proof of concept.

References

[1] M.F. Goodchild, M. Yuan, and T.J. Cova. Towards a general theory of geographic representation in gis. International Journal of Geographical Information Science, 21(3):239–260, 2007.